
Title: Mining Twitter for Crisis Management: Real-time Floods Detection in the Arabian Peninsula

Name: Waleed Alabbas

# Mining Twitter for Crisis Management: Real-time Floods Detection in the Arabian Peninsula

By

Waleed Alabbas

A thesis submitted to the University of Bedfordshire in partial fulfilment of the requirements for the degree of Doctor of Philosophy

University of Bedfordshire

Institute for Research in Applicable Computing

April 2018

I Waleed Alabbas declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

Mining Twitter for Crisis Management: Real-time Floods Detection in the Arabian Peninsula

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have cited the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Either none of this work has been published before submission, or parts of this work have been published as indicated on [insert page number or heading]:

Name of candidate: Waleed Alabbas Signature:

Date: 06/04/2018

## Abstract

In recent years, large amounts of data have been made available on microblog platforms such as Twitter, however, it is difficult to filter and extract information and knowledge from such data because of the high volume, including noisy data. On Twitter, the general public are able to report real-world events such as floods in real time, and act as social sensors. Consequently, it is beneficial to have a method that can detect flood events automatically in real time to help governmental authorities, such as crisis management authorities, to detect the event and make decisions during the early stages of the event.

This thesis proposes a real time flood detection system by mining Arabic Tweets using machine learning and data mining techniques. The proposed system comprises five main components: data collection, pre-processing, flooding event extract, location inferring, location named entity link, and flooding event visualisation. An effective method of flood detection from Arabic tweets is presented and evaluated by using supervised learning techniques. Furthermore, this work presents a location named entity inferring method based on the Learning to Search method, the results show that the proposed method outperformed the existing systems with significantly higher accuracy in tasks of inferring flood locations from tweets which are written in colloquial Arabic. For the location named entity link, a method has been designed by utilising Google API services as a knowledge base to extract accurate geocode coordinates that are associated with location named entities mentioned in tweets. The results show that the proposed location link method locate 56.8% of tweets with a distance range of 0 – 10 km from the actual location. Further analysis has shown that the accuracy in locating tweets in an actual city and region are 78.9% and 84.2% respectively.

# Table of Contents

# List of Tables

# List of Figures

## List of Abbreviations

| Abbreviation | Term |
|---|---|
| AU | Active users on social media. |
| AI | Artificial Intelligence |
| API | Application programming interface. |
| C5.0 | Decision-tree. |
| CA | Classic Arabic. |
| CHI | Chi-square |
| DA | Dialectal Arabic. |
| DRT | Dimensionality Reduction Techniques |
| DSS | Decision Support Systems. |
| FS | Features Selection. |
| IOB | Inside I, Outside O and Beginning B tagging scheme |
| IR | Information retrieval. |
| CRF | conditional Random Field |
| K-NN | K-Nearest Neighbour. |
| L2S | Learning to Search |
| lat | latitude coordinate |
| long | longitude coordinate |
| ML | Machine learning. |
| MSA | Modern standard Arabic. |
| NA | not available |
| NB | Naïve Bayes. |
| NE | Named Entity |
| NED | New Event Detection. |
| NEL | Named Entity Linking |

| | |
|---|---|
| NER | named entity recognition |
| NLP | Natural Language Processing. |
| NNET | Neural Networks |
| NOAA | National Oceanic and Atmospheric Administration |
| OSM | Open Street Map |
| OSN | Online social network. |
| POI | *point-of interest* |
| POS | Part-of-speech. |
| RED | Retrospective Event Detection. |
| SLR | Systematic Literature Review. |
| ST | Stemming Techniques |
| SVM | Support Vector Machines. |
| TC | Text classification. |
| TF-IDF | Term Frequency-Inverse Document Frequency |
| TW | Term Weighting. |
| UGC | User-generated content. |

## List of Publications

The following refereed publications are part of the key results from the research work presented in this thesis, which has previously been published in international conferences.

- Alabbas, W., Al-Khateeb, H.M. and Mansour, A., 2016, October. Arabic text classification methods: Systematic literature review of primary studies. In Information Science and Technology (CiSt), 2016 4th IEEE International Colloquium on (pp. 361-367). IEEE.
- Alabbas, W., al-Khateeb, H.M., Mansour, A., Epiphaniou, G. and Frommholz, I., 2017, June. Classification of colloquial Arabic tweets in real-time to detect high-risk floods. In Social Media, Wearable and Web Analytics (Social Media), 2017 International Conference On (pp. 1-8). IEEE.

# Chapter 1: Introduction

Social media platforms are becoming increasingly popular. This can be evidenced by the huge and increasing amount of text generated by users on a day-to-day basis. According to Smith (2017) there were 2.46 billion active users (AU) on social media platforms in 2017. The same report estimates that the number of AU on social media platforms will increase by 18% to 2.9 billion AU in 2020. Moreover, this translates to the availability of huge data and a wealth of information (Medvet and Bartoli, 2012). Meanwhile, the availability of this huge data introduces many challenges for researchers who are trying to analyse and process such data.

Twitter is an online social media platform that allows users to post short texts called tweets with the option to attach a link to a website, photograph, or video. The popularity and features of Twitter have paved the way for practical applications to consider real-world events. For instance, Sakaki et al., (2010) developed an earthquake reporting system by analysing Japanese tweets. Their system detects earthquakes faster than the Japan Meteorological Agency. Events detection generally works by scanning Twitter for texts, or so-called tweets, containing keywords relating to a defined topic or theme, which in turn, can be compiled and analysed for newsgathering, research, intelligence, and the detection of disasters and other important occurrences.

Twitter users generated 500 million tweets per day on average in 2016, and just over one-third (34%) of all tweets were in the English language, with 6% in Arabic, making it the sixth most used language on Twitter. Furthermore, Aslam, (2017) report that Saudi Arabia has the highest number of AU on Twitter.

In this thesis, an effective method for detecting flood events in real time, by collecting and analysing tweets which are written in Arabic, is presented. Furthermore, a flood detection system that employs tweets and rainfall data from reliable global organisations has been developed to help emergency authorities to detect and track flood events and their locations.

In this chapter, the research introduction will be discussed, including the background, motivation, main contributions and the organisation of this thesis.

## 1.1. Background

Currently, existing User-Generated Content (UGC) platforms such as Twitter are being considered as powerful channels for sharing and exchanging information over the Internet. In the event of a natural disaster or other event, Twitter users generate a huge number of messages sharing and discussing event information and their opinions on this event. The tweets' content can lead to detecting significant events in the Twitter stream. However, to properly detect events on social media platforms, it is necessary to define and understand the concept of social media platforms and how users are reporting events using such channels.

The following sub-section, firstly addresses how users propagate events-related contents on Twitter. Secondly, it will define the concept of event detection from an emergence management perspective, consequently categorising event detection on social media into four types based on event type and detection task. Thirdly, characterising and distinguishing between Arabic and other Languages will be discussed. Last but not least, the limitations of the existing work on event detection from social media in general, and event detection from Arabic tweets, will be discussed.

### 1.1.1. Message Propagation on Social Networks

Social media platforms are highly interactive platforms via which their users publish, discuss and post thousands of user-generated content, including a wide range of topics and events. In 2007 Ellison (2007) defined social media networks sites as “web-based services that allow individuals to construct a public or semi-public profile within a bounded system, articulate a list of other users with whom they share a connection, and view and traverse their list of connections and those made by others within the system”. Later in 2008 Krishnamurthy and Dou (2008) defined social media sites as user-generated services whereby users can produce, design, publish, or edit content such as blogs, microblogs, online review sites, virtual game worlds, images or video sharing sites.

In the 1970s, Dynes (1970) studied victims of disaster behaviour and observed that they do not lose control and make quick decisions based on the information available to them at the time. The first responders to a disaster event are often neighbours and friends and other members of the public. They rush to the scene to perform search and rescue operations, administer first aid, and perform critical tasks necessary during the first moments of response.

The growing use of social media to gather and share information, organise relief efforts, and communicate nowadays, means that the members of the public who can play a valuable role in these situations is no longer limited to those in the areas of impact. The following scenario summarises how social media users and emergency authorities react during significant events (Latonero and Shklovski, 2011):

1) In the event of a natural disaster or significant event, Twitter users who observed this event post tweets about the event to share event information with their followers.

2) In a short time, Twitter users retweet tweets that are received from their social networks and share the event information which comes from traditional media and official sources.
3) Emergency Management Agencies employ Twitter and other social media platforms to share event information with the public, especially affected communities.
4) After that, Emergency Management Agencies monitor Twitter data streaming to gather information during the time of the event.

This thesis focuses on detecting real-time events on Twitter, and a set of Twitter specific terms is used throughout. Table 1-1 shows the most popular twitter terms and their definitions.

In terms of social connectivity, Twitter allows a user to follow any number of other users. The Twitter user can follow other users without requiring approval. Users can set their privacy preferences so that their updates are available only to each user's followers. By default, the posted messages are publicly available to anyone. In this work, only messages posted publicly on Twitter are considered.

Twitter provides an application programming interface (API), which allows developers to programmatically access the public data streams as well as many features of the service. For instance, Twitter streaming API provides filtering by location, keywords, author, and so on. The availability of Twitter data has motivated significant research work in various disciplines and led to numerous applications and tools.

*Table 1-1 Twitter term definition (Twitter, 2018b)*

| Term | Definition |
|---|---|
| Twitter | An information network made up of 280-character messages (including photos, videos and links) from all over the world. |
| Tweet | A Tweet may contain photos, videos, links and up to 280 characters of text. |
| @username | A username is how you're identified on Twitter, and is always preceded immediately by the @ symbol. |
| Retweet (RT) | A retweet is a repost or forward of a tweet by another user. |
| Like | Likes are used by users when they like a tweet. By favoriting a tweet, a user can let the original poster know that they have liked their tweet. The total number of times a tweet has been favorited is visible to everyone. |
| Followers | A follower is another Twitter account that has followed you to receive your Tweets on their Home timeline. |
| Followees | People who the user followers. The total number of followees a user has is also visible to everyone. |
| hashtag | Written with a # symbol—is used to index keywords or topics on Twitter. This function was created on Twitter, and allows people to easily follow topics they are interested in |
| reply | A reply is a response to another person's Tweet. |
| mention | A mention is a Tweet that contains another person's @username anywhere in the body of the Tweet. |
| User profile | The user profile displays user information, as well as all Tweets that have been posted. |
| monthly active users | Is someone who logs in (but does not necessarily tweet) once a month. |

### 1.1.2. Arabic Text

Arabic is the fifth most widely used language in the world. It is officially used in 22 countries, and is the mother tongue of more than 422 million persons and the second language

of almost another 250 million. Arabic has 28 letters and the orientation of writing is from right to left. Its script has a unique shape, marks, diacritics, style (font), numerals, distinctive letters and none distinctive letters (Odeh et al., 2015). Noaman and Al-Ghuribi (2012) have discussed its complex morphology and how words could have different meanings within a given context. Arabic is highly inflectional and derivational (El-Halees, 2007); it does not use capitalisation for proper nouns, which is a very useful input when classifying English documents. Arabic synonyms are widespread (Saad and Ashour, 2010). The majority of words have a tri-letter root, while the rest have a quad-letter root, penta-letter root or hexa-letter root (Khreisat, 2009).

Arabic can be classified with respect to its morphology, syntax, and lexical combinations into three different categories: classic Arabic (CA), modern standard Arabic (MSA), and dialectal Arabic (DA) (Habash, 2010). There are many varieties of Arabic dialects distributed across the Arab world. There are often several variants of a dialect within the same country. Colloquial Arabic text is a collective term for the spoken languages or dialects of people throughout the Arab world, and is therefore widely used on social media. It is generally characterised as being highly unstructured, inconsistent, and difficult to process. In natural language processing, researchers have classified Arabic dialects into five different groups, namely: Egyptian (spoken in Egypt, but understood universally), Maghrebi (spoken in all of North Africa), Gulf (spoken primarily in Saudi Arabia, UAE, Kuwait and Qatar), Iraqi (spoken in Iraq), and Levantine (spoken primarily in the Levant, Syria and Palestine) (Cotterell and Callison-Burch, 2014) (Al-Sabbagh and Girju, 2012). In contrast, MSA is the official Arabic language that is taught in schools and defined as the accepted medium for formal communication. MSA is therefore more understood, has specific grammar rules, and is structured and more consistent.

Arabic is a morphologically rich language as a result of the following points:

- A given term can take on several meanings depending on the context, such as (ذهب) which may mean *gold* or *went*.
- Many people tend to use the dialect of their country or English words in Arabic letters instead of using MSA such as (أشوف= أرى) or ( Coffee = كوفي )
- Repeating the letter more than once to intensify the meaning or feeling (which can be seen clearly in social media) such as جمييييل, which in MSA is written as جميل.
- Arabic has various diacritics; diacritics change the meaning of words. For example, (“confusing” “لَبْس”) and (“dress” “لِبس ”) can both be read as the same word when written without diacritics.

Recently, Twitter has been growing exceptionally fast in the Arab world. The Arabic Social Media Report 2017 (Salem, 2017) states that there are 11.1 million monthly active users in the Arab world, generating 27.4 million tweets per day, and 72% of those tweets are written in Arabic. More than one third of the tweets generated in the Arab world come from Saudi Arabia.

Users on social networks typically use colloquial Arabic by using their local dialect such as Egyptian and Gulf dialects (Al-Sabbagh and Girju, 2012). In natural language processing (NLP), dealing with text written in a colloquial language creates additional challenges, and these challenges can clearly be seen when moving to microblog platforms such as Twitter.

### 1.1.3. Event Definition and Categorisation

In general, an event can be defined as follows:

***Definition 1.1***. *An “event” is something that occurs in a certain place during a particular interval of time (Dictionary, 2018).*

Events on social media platforms can be loosely defined as real-world happenings that occur within similar time periods and geographical locations, and that have been mentioned by online users in the forms of images, videos or texts. A slightly different definition that also includes the concept of event on social media is given by (Dong et al., 2015). These authors identify an "event" on social media as follows:

***Definition 1.2.*** Events on social media are real world happenings that are reflected by data that are concentrated either in both time and space, or in at least one of these two dimensions.

Atefeh and Khreich (Atefeh and Khreich, 2015) have presented a survey of techniques that can be applied to Twitter-based event detection. They have classified event types as 'specified' or 'unspecified'; detection tasks as 'Retrospective Event Detection (RED)' or 'New Event Detection (NED)', and detection methods as 'supervised' or 'unsupervised' methods.

***Unspecified Event Detection***: The nature of Twitter posts is that they reflect events as they unfold; hence, these tweets are particularly useful for unknown event detection. Unknown events of interest are typically driven by emerging events, breaking news, and general topics that attract the attention of a large number of Twitter users.

***Specified Event Detection***: Specified event detection includes known or planned social events. These events could be partially or fully specified by the related content or metadata information, such as location, time, venue, and performers.

***New event detection (NED):*** NED involves the discovery of new events from live streams in (near) real time. NED techniques involve continuous monitoring of Twitter signals to discover new events in near real time. They are naturally suited to detecting unknown real world events or breaking news. Although NED approaches do not impose any assumptions on the event, they are not restricted to unspecified event detection. When the monitoring task

involves specific events (natural disasters, celebrities, etc.) or specific information about the event description (e.g., geographical location), this information could be integrated into the NED system.

***Retrospective event detection (RED):*** focuses on discovering previously unidentified events from accumulated historical collections.

## 1.2. Limitations of Existing Work and Challenges

This section highlights a number of challenges met by researchers when dealing with event detection from social media.

***Volume and Velocity***: The volume and velocity of social media messages posted during crises are extremely high, which makes it hard for decision makers to respond in a timely manner.

***Real-Time Event Detection***: flood events should be identified as soon as possible. In this case, methods for flood detection should consider the process time to detect flood events.

***Noise and Veracity:*** social media platform user generated information is characterised by noise. Noise in social media flows from advertisements, spam messages and bot accounts that publish large volumes of messages. Another obstacle is that textual information on social media is limited. Users usually publish very short messages, which makes it difficult to apply text mining and NLP methods.

***Machine learning (ML) approaches:*** In ML selecting, the most suitable approach to apply in supervised learning is not a trivial task. Textual representations such as Term-Document matrices are not sufficient. For event detection, the ML approach should be based on the content-based attributes that appear in messages.

***Evaluation:*** The most important task in ML is evaluation, which compares the result produced with other existing studies' results by using open datasets. Unfortunately, the availability of

event detection datasets is very limited, and none of these data sets are suitable for this work, as it focuses on a specific type of event (floods) and specific language (Arabic).

Named entity recognition (NER) is an important form of Information Extraction (IE) task, which identifies snippets of a text that mention events and issues in the real world, and it is used for a number of IE tasks, such as NEL, co-reference resolution, and relation extraction. However, NER is often difficult to utilise with user-generated content, especially within the microblog genre, due to limited amount of contextual information contained within short messages, as well as the limited curation of content conducted by third parties, including editors for newswire. This section will highlight various state-of-the-art NER methods and their performance with regard to microblog data.

NER systems use a number of techniques, such as gazetteer-based lookups, for example the ANNIE system produced by GATE (Cunningham et al., 2002); DBpedia Spotlight (DBpedia, 2017); Lupedia (Ontotext, 2018); machine learning-based methods for detecting named entities (e.g. the Stanford NER system) (Finkel et al., 2005); TextRazor (Txtrazor, 2018), and Alchemy API (IBM, 2018). It is only recently that Twitter has been considered an active research topic for NER. Moreover, various approaches have been suggested for its application in the English language, for example, the approach by (Ritter et al., 2011), which involves applying tokenisation, along with POS tagging and topic models in order to look for named entities. Importantly the system put forward by Ritter et al (2011) has been shown to perform better than the Stanford NER system with Twitter datasets.

NER systems may find it hard to cope with colloquial Arabic text due to the ambiguity of some words that are undiacritized, which results in them having different meanings

according to the specific context they are used as a result of the missing diacritics, and this makes NER Systems even more complex than usual (Abdallah et al., 2012). The majority of modern Arabic written texts are written in colloquial Arabic, but previous work has addressed NER only with Modern Standard Arabic (MSA) text, such as news articles (Zayed and El-Beltagy, 2012); (Zirikly and Diab, 2014). Therefore, (Abdallah et al., 2012) have put forward a simple integration method that combines a rule-based system and a machine-learning classifier to better recognise Arabic entries. To do so, (Abdallah et al., 2012) repeated the work by (Benajiba et al., 2008) and integrated it with a Decision Tree classifier, which resulted in an integrated system that achieved 8-12% better than the rule based system alone.

A NER system designed for colloquial Arabic, specifically the Egyptian Dialect, was introduced by (Zirikly and Diab, 2014), which could identify person and location named entities by applying: Inside I, Outside O and Beginning B (IOB) as the tagging scheme. To test the system, they chose an Egyptian dialect dataset from a set of web blogs made up of almost 40k tokens that included 285 people and 153 location named entities. (Zirikly and Diab, 2014) found that their system reached an F-score of 91% for location NER when they applied a Levenshtein match approach to discover the similarities between the input and a gazetteer entry. (Zirikly and Diab, 2015) used two datasets in order to develop an Arabic entity system- one containing Arabic tweets, and the other an Egyptian dialectal dataset. They found that the performance of the Arabic NER system proposed underperformed when tested with a Twitter corpus, and the highest F-score result achieved was 61% when applying gazetteers; Brown Clustering (Brown et al., 1992), and Word2vec Clustering (Mikolov et al., 2013). Yet their system achieved an F-score of 91% for location NER when a formal Egyptian dialectal dataset was used.

As well as addressing the location NER task, the current research has explored inferring the event location through tweets that mention flood events and contain more than one location NE. The aim of this was to infer a single location that refers to the specific flood event location.

There are many existing flood event detection systems which have been developed by researchers or organisations. The following list shows the limitations of the existing systems:

- Does not support the Arabic language (petabencana, 2018, Tag, 2018). Most existing systems focus on the English Language because of the difficulty in analysing text written in other languages (e.g. Arabic, Persian).
- Does not utilise trusted and real-time rainfall data which could help to analyse and detect flood events (petabencana, 2018; Tag, 2018).

## 1.3. Motivation and Research objectives

Social media platforms are important sources for spreading real time information about natural disaster events (such as floods, earthquakes and tornadoes) and significant events like traffic jams, fires, riots and terrorist attacks. The huge amount of information has inspired researchers to detect and track those events. Natural disaster events need to be detected in real time in their early stages when they are observed by social media users. However, with such short and noisy messages coming from social media platforms, it is difficult to detect the event by filtering and sorting posts manually. Consequently, it would be beneficial to have a method that can perform this task automatically in real time to help governmental authorities to detect the event and make decisions during the early stages of the event. Recently, floods have become the most frequent natural disaster in the Arabian Peninsula; they occur several times a year and cause severe threat and damage to life and

property. In one of the worse cases, in 2009, over 120 civilians were reported to have been killed. Part of the reason for such severe damage was the lack of real-time information on the event, and inefficient Decision Support Systems (DSS) to utilise available resources efficiently (Mashael Al, 2010); (Al-Saggaf, 2012).

Many researchers have proposed models and techniques for the purpose of detecting natural disaster events on social media, but these have mostly focused on English as the communication medium, whereas other languages have received relatively less coverage. As geography and NLP tools all differ in their language, this work is important in its own right for location inference with the Arabic language. Moreover, to the best of our knowledge, and further to the very limited amount of research on event detection from Arabic users in general, there is no evident research that has addressed flood detection from Arabic tweets.

The objective of the current research is to design a system that can be used for flood events detection in real time. The system will be implemented, tested and analysed to check its accuracy and viability. This main objective has been divided into the following two sub-objectives:

*Table 1-2 sub-objectives*

| # | ***Sub-objective*** |
|---|---|
| O1 | Design and implement a real-time detection system for flooding |
| O2 | Analyse the performance of this aforementioned system. |

- $\mathbf{0_1}$**:** The design and implementation of a real-time detection system for flooding will show that data from Twitter users can be used to create sensors for detect flooding events in real time.

- **$0_2$**: An analysis of the performance of the system will help in refining the algorithm by considering parameter selection, and carrying out further testing and analysis to discover its limits.

## 1.4. Problem statement and research questions

All emergency events require a fast response, and decisions based on first-hand information. Short messages posted on social media sites such as Twitter can typically reflect these events as they happen, because people often use Twitter to report real-life events, which can improve the information basis for disaster response with regard to natural disasters. The impact of natural disasters appears to be worse due to not identifying floods in real time, as well as the lack of mitigation implementation and slow response rate.

| **Floods Impact** | | **Probability** | | | | |
|---|---|---|---|---|---|---|
| | | Highly unlikely | Unlikely | Possible | Likely | Very likely |
| | | **1** | **2** | **3** | **4** | **5** |
| Catastrophic | **5** | | | | | |
| severe | **4** | | | | **High Risk** | |
| Major | **3** | | **Medium Risk** | | | |
| Minor | **2** | | | | | |
| Low | **1** | **Low Risk** | | | | |

*Figure 1-1 Flood Risk Matrix*

Floods are natural, yet they are notorious disasters since they have the potential to occur suddenly (e.g. breakdown of a dam or quick melting of snowfall). However, crisis management works to minimise the level of damage they cause to people, places and communities by identifying them in real time. This work has applied a Flood Risk Matrix, as shown in Figure 1-1 to determine levels of flood risk. The matrix works by adding the ratings for the 'likelihood' of flooding to the ratings for predicted 'consequences', giving an overall risk score. This risk score is then used to rank the flood risk. This risk ranking is then used to help classify each tweet into a positive class (tweets mentioning high risk floods) or negative class (otherwise).

Humanitarian organisations and volunteer networks often set up live online, manual crisis mapping sites during natural disaster events, for example, the Haiti earthquake of 2010; Hurricane Sandy in 2012, and the Oklahoma tornado in 2013. In this way, they take crowdsourced information from news reports, social media, and civil protection agency alerts, and check and filter it before presenting it live on online crisis maps made available to the general public to view. However, there are a number of challenges in doing so, such as automating the task of carrying out real-time data fusion involving huge volumes of heterogeneous information from multiple sources and ensuring this data is trustworthy and credible.

Real-time geospatial information systems (GIS) usually map social media microblog reports, and they typically use geotag metadata with longitude/latitude coordinates. In this way, social media data can be turned into a crowdsourcing virtual sensor network, including mapping Twitter messages ready to be plotted. The US Geological Survey (USGS) explains that a major advantage of Twitter-based detection systems, rather than sensor-based systems, is their speed in detection speed and low costs. It is possible to combine social media GIS systems

with conventional GIS systems to deploy hardware-based sensors, for example in situ seismic sensors; remote sensing aerial photography, and satellite imaging. This will lead to forming a coherent picture of what is going on, and it can be presented to emergency responders; civil protection authorities, and even the general public, in order to better coordinate response efforts and create greater awareness.

A disadvantage is that just one percent of all tweets contain geotag metadata, and out of these, the geotags can be a combination of genuine mobile devices (using GPS) and Twitter's default information on the user's home location. The tweeted text may also contain references to additional locations that are geospatially far from the location of the device that has sent the tweet. While this may not be a problem for mapping course-grained earthquake regions, it presents problems for finer-grained maps that are required for monitoring flooding.

(Amezquita-Sanchez et al., 2017) carried out s state of the art review on how different flood prediction systems work, and they found that there are different types and sources of data need to be made use of by applying big data technologies in order to further develop and improve flood prediction models. In addition, they claim that rainfall data collected via satellite imagery and monitoring stations are important sources of data for predicting flood events.

In their review of flood prediction technologies (Amezquita-Sanchez et al., 2017) discovered the importance of using a range of data sources, and that there is a gap in the knowledge, therefore, they suggest attempting to incorporate rainfall data, including weather station,

satellite data and social media data, as a way of conducting further research on developing flood prediction systems.

Event detection systems have been applied to detect events from social media networks. Multiple authors (de Bruijn et al., 2018, Jongman et al., 2015, Eilander et al., 2016) have developed flood detection systems by analysing social media networks' content, although none of those systems used the Arabic language. Furthermore, none of those systems studied or analysed rainfall data. This work focuses on filling this gap by developing a flood detection system that includes Arabic text. Moreover, the proposed system analyses rainfall data collected from a reliable global organisation (National Oceanic and Atmospheric Administration).

In summary, this research aims to address the gaps in flood detection by developing a flood detection system that can be achieved by answering the following research questions:

- RQ1: How is it possible to identify and extract particular events from social media networks' messages where the writing used contains the form of a short description or keyword tags? (Abbreviations are also widely used in a message; moreover, the messages are often noisy).
- RQ2: How events' locations be inferred and extracted from social media messages?
- RQ3: How can an inferred location be linked to maps to produce event visualisation and information?
- RQ4: How can weather data be used and integrated with social media data?

## 1.5. Scope and Limitations

The main focus of this work is to develop a flood detection system by extracting flood events in real time from Twitter using machine learning (ML) techniques, and provide information about detected events to help emergency authorities to detect and track flooding events. The information collected on events can either be used for social awareness, or to help decision makers within emergency authorities. In this research, the Twitter dataset collected from Twitter Streaming API has been analysed. The scope of the research is restricted to analysing only this specific event in the Twitter dataset. Furthermore, because of using supervised learning techniques in the proposed system, the scope is restricted to tweets written in the Arabic language. The most important requirement of a natural disaster detection system is that it can be used in real time monitoring; therefore, it is essential that this system focuses on the effectiveness and efficiency of detecting the event in real time.

## 1.6. Significance and contribution of the research

This research focuses on proposing a system for detecting flood event from social media platforms, especially Twitter, using ML and data mining techniques. The major contributions of this research are outlined as follows:

- A Systematic Literature Review (SLR) utilising a strict search protocol to summarise the key characteristics of the different Text Classification (TC) techniques and methods used to classify Arabic text. This SLR focuses on critical analysis of the literature from year 2006 to 2014 which will serve the scholars and researchers to

analyse the latest work of Arabic text classification as well as provide them a baseline for future trends and comparisons.

- An effective method to classify natural events from short messages posted on Twitter is proposed. In order to classify event-related tweets and achieve high performance accuracy, we study the impact of light stemming and removing prefixes and suffixes on colloquial Arabic text classification. We compare Naive Bayes (NB), Support Vector Machines (SVM), Decision-tree (J48, C5.0), Neural Networks (NNET) and K-Nearest Neighbor (k-NN) classification algorithms, showing that using light stemming or removing prefixes and suffixes contributes to reducing classification performance in term of accuracy
- A location inferring method is incorporated to extract events' locations, as mentioned in social media messages. Identify events locations from tweets is one of the main challenging tasks related to social media currently addressed in the NLP community. Existing NER methods have mainly focused on extract all mentioned location NE and do not distinguish the event location NE from other location NEs which are mentioned in a tweet. Furthermore the state of the art NER Systems based on gazetteers or conditional Random Field (CRF) models underperformed when tested on the Twitter corpus for both the Arabic and English languages (Zirikly and Diab, 2015); (Derczynski et al., 2015). As a third and main contribution, we propose a novel method based on the learning-to-search (L2S) approach that identify event location NE mentions and distinguish it from other location NEs mentions in tweets and outperform existing Arabic NER systems in term of accuracy when tested on colloquial Arabic text.

- An effective Named Entity Linked (NEL) method to addresses the geo/non-geo or geo/geo ambiguity which occur when more than one place has the same name entity but are actually different spatial locations. We have developed algorithm by utilising Google maps API to identify the most accurate location associated with the event location NE in tweets.
- Implementing a flood detection system that provides users with an overview of live floods events. The system supports event tracking by allowing users to specify the time period and search keywords in order to visualise the floods events on Google maps. Furthermore, the system visualises live rainfall amounts on the same maps.

## 1.7. Thesis Organisation

In this thesis, the problem of flooding event detection in real time is investigated, and solutions are provided to address this problem. Techniques are presented that are related to three main areas: Arabic text classification, location Named Entity (NE) and Named Entity Linking (NEL). This thesis is organised into seven chapters, as summarised below:

- Chapter 1 - Introduction

This chapter presents a brief background and provides important definitions for the research. Furthermore, the motivation, problem statement and the scope and limitations of the research are stated. The chapter concludes with the contributions of this work.

- Chapter 2 – Background

This chapter sets out the machine learning concepts, definitions and algorithms used in this research. The chapter contains five sections, which are as follows: the first and second

sections describe the classification of texts and their architecture. The third section lists and defines the classifiers used in the experiments in this research. The fourth section presents the performance measures that have been used in the research, and finally, the fifth chapter describes the Learn to Search (L2S) method that has been used for the NER task in this research.

- Chapter 3 - Literature Review

The research works related to this thesis that have been produced in recent years are reviewed. The chapter is divided into two main sections: The first section covers the work that has been done on Arabic text classification, through a systematic review that covers this area. The second section discusses what works have been produced for event detection from social networks. This chapter also presents various NER techniques and focuses on extracting location NER from Arabic text.

- Chapter 4 – Research Methodology

The methodology used in the current research is explained in this chapter, beginning with the rationale behind the choice of the research methods. A flow chart is included, which illustrates the methodology and the processes followed in the research, and then the specific methods used are explained in more detail. The chapter ends by outlining how the experiment has been conducted and the process of analysis, followed by a summary of the chapter's main content.

- Chapter 5 - Classification of Colloquial Arabic Tweets in real-time to detect high-risk floods

A set of different experiments are proposed to study the classification task in colloquial Arabic text. In particular, the impact of the stemming process on Colloquial Arabic text is

investigated. In this chapter, machine learning methods are used to separate flood event content from non-flood events using mainly the Support Vector Machine (SVM), Naïve Bayes (NB), Neural Networks (NNET), K-Nearest Neighbor (k-NN), Decision Tree j48 and C5.0 classifiers. The chapter concludes with a discussion of the experiments' results.

- Chapter 6 - Location Inference from Twitter

The types of location and spatial features on Twitter are identified and the spatial features used for flood event detection are stated. Then the Learning to Search (L2S) method is introduced to be used to infer flood event location's NE. In addition, the Location NER method is presented to infer the event location from Arabic tweets and share the proposed method's results. The chapter concludes with a comparison of the proposed method results with existing Arabic NER systems.

- Chapter 7 - Location Named Entity Linking

This chapter introduces and defines location Named Entity Linking (NEL) tasks and lists the challenges faced in solving location NEL using colloquial Arabic text. The approach to linking the location NE to an instance in a knowledge-base is explained. The chapter concludes with an evaluation of the proposed approach and the results.

- Chapter 8 - Real-time Flood Detection system

This chapter presents the real-time floods detection system by combining tweets data which is extracted from Twitter, and rainfall data which is extracted from National Oceanic and Atmospheric Administration (NOAA). The system structure and implementation screenshots and system limitations are presented and explained.

- Chapter 9 - Conclusions and future work

The main conclusions and contributions of the work are summarised in this chapter. The practical outcomes of this research for flood detection systems, and the outlook for the future direction of the work are highlighted.

# Chapter 2: Background

## 2.1. A mathematical definition of the text classification task

Text Categorisation is the discipline within which the automatic classification of text documents under predefined categories or classes is carried out (Wright et al., 1999). The task of text categorisation comes under the Automatic Classification, which may be referred to as Pattern Recognition, and used within Machine Learning. Classification is important, and if unsupervised classification is used, the possible final classes cannot be predicted. However, unsupervised classification techniques may be useful for discovering possible groups or classes, such as Document Clustering. Where the class or classes are predefined, supervised techniques can be used to find an approximate solution to a problem, provided that a training collection is available, and this is usually referred to as Text Categorization, with Document Clustering coming under a different discipline.

It is possible to identify three paradigms within text categorisation as illustrated in Figure 2-1 to binary cases, multi-class cases, and multi-label cases.

- For binary cases, the sample will belong to one of two given classes, and so the classifier must determine which of these two the sample goes with.
- For multi-class cases, the sample belongs to a specific class of m classes.
- For multi-label cases, the sample can belongs to a number of classes at the same time, as classes could overlap through the documents used.

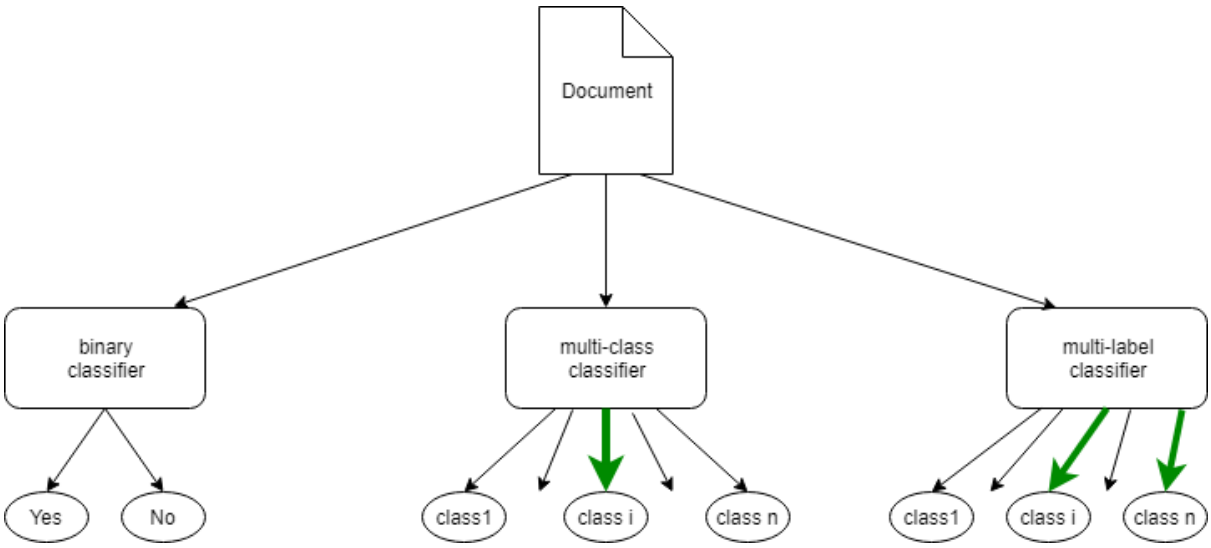


*Figure 2-1 Paradigms in text classification*

For binary classification, supervised algorithms are used to train a classifier, and a sample document is assigned to one out of two potential sets. These potential sets are normally labelled belonging samples, which are positive, or not belonging samples, which are negative respectively; or as two different classes in a one against one approach to classifying. Binary classification allows a number of different algorithms to be used, such as Linear Regression; Naive Bayes; Support Vector Machines (SVM). The binary case forms the initial case, and the two other cases can be built based on this. For multi-class or multi-label tasks, a binary classifier is usually trained for every class according to the traditional approach, with the binary base case revealing the level of confidence in the classification, whether it is ranked first (multi-class assignment),or is a top ranking one (multi-label assignment). Researchers such as Allwein (Allwein et al., 2000) have addressed these three paradigms and have come up with a common framework that can help to reduce multi-class problems and switch t margin-based classifiers to a binary approach.

## 2.2. Architecture for a text categorization system

The scheme contains several variants, although they involve only minor differences to the common design. The following components can be used to build a text categorization system (Montejo Ráez, 2005):

1. Feature extraction: The text is in plain format (ASCII) and is processed and indexed according to the following two steps:

    a) Feature identification, which involves identifying which features to keep, such as words, entities, bigrams or stemmed terms that are representative components of the original document. Feature identification usually involves techniques relevant to natural language processing and the retrieval of statistical information.

    b) Feature weighting, as after the features have been selected, it is necessary to calculate a weight for measuring whether they are relevant to the document, and for this; a number of approaches may be used to do this, although the bag-of-words approach is usually used as it is a traditional way of simplifying the processing.

2. Dimensionality reduction (DR): Retrieving information and text categorisation often result in problems due to the great number of distinctive features involved, and because every word in piece of text could be a required feature. While stemming and other lemmatization techniques could lessen this number, it still presents an issue when using Machine Learning (ML)

techniques to for text categorisation, and so alternative methods are suggested so as to reduce the dimensionality of the set of features. This includes ranking the ability of a term concerning whether it is a good indicator for documents as well as classes. In this way, useful information for discarding terms can be provided by using measures such as Chi square, mutual information, information gain, or even document frequency (Yang and Pedersen, 1997). Through term selection, or discarding, it is possible to reduce the dimensionality of the feature space; alternatively, term extraction, or feature transformation, could be used. This requires replacing words with other entities that cover the existing stems or words, and grouping them together to form semantic sets based on a unique global characteristic. Feature extraction and dimensionality reduction can therefore transform text documents into samples ready for use in the future with learning algorithms.

3. Classifier training: This forms the core of the system and involves using machine learning methods to create an autonomous classifier through the use of supervised learning algorithms. The wide variety of classification approaches (e.g. statistical, probabilistic, neural networks, fuzzy logic etc) used for a range of problems, in particular, pattern recognition and its variants, means that the vast range of algorithms available can be overwhelming. However, the research on comparing different classifiers for text categorisation, highlights the most useful potential approaches, including Naive Bayes (McCallum and Nigam, 1998), Support Vector Machines (Sun et al., 2009). The conversion of the documents into feature lists in the previous

stage manes that the algorithms should work in the same way as for other types of data in text format.

4. Thresholding: In order to reach a hard classification involving yes or no answers, in addition to classifier outputs that have a value that matches a document and a class, it must be decided whether the document should be assigned to a category, and this is done by assigning a specific threshold. The threshold is fixed, and the value forms a decision boundary, when it comes to the S-Cut, R-Cut or P-Cut approaches (Yang, 2001). Alternatively, the limit may be based not on the classification status value, but on the number of classes assigned to a document. This means that a fixed number of classes will be attached to every document (Pouliquen et al., 2006).

In summary, it is necessary to convert a document according to a specific set of features that the classification algorithm can use. The learning process is dependent on the algorithms used as the proposed architecture is based on these. Usually, a training collection divided into a learning set and a validation set is used.

## 2.3. Classifiers

So far it has been shown how documents can be converted by forming a list of specific attributes or features for condensing down the original text. In this way documents can be changed into a format that is more suitable for learning algorithms. This is useful due to the extant amount of machine learning research that has led to a

huge choice of supervised algorithms for training binary classifiers. The options are so wide ranging that a detailed description of each one is not possible, therefore only those relevant to the current system will be explored, along with some brief reference to a few others.

### 2.3.1. Support Vector Machines

Support vector machines use the Structural Risk Minimization principle of computational learning theory, which involves finding a hypothesis *h* for ensuring the lowest true error (Joachims, 1998). The true error of *h* refers to the probability of *h* making an error with an unseen and randomly selected test example. Furthermore, it is possible to set an upper boundary to connect the true error of a hypothesis h with the error of h for a training set using VC-Dimension, and the complexity of the hypothesis space containing h. Support vector machines can be used to discover the hypothesis h, which will help to reduce the boundary of the true error by controlling the VC-Dimension of H.

SVMs are universal learners, and in their basic form they can learn a linear threshold function. In addition, the simple “plug-in" of the correct kernel function means they can be used to learn: radial basic function (RBF) networks; polynomial classifiers, and three-layer sigmoid neural nets (Joachims, 1998).

A very useful property of SVMs is their ability to learn independently of the dimensionality of the feature space, as they measure the complexity of hypotheses according to the margin set for separating the data, rather than the number of features, which allows them to generalise despite perhaps a high number of features, provided the data is separated by a wide enough margin using hypothesis space functions. This feature regarding margins facilitates excellent parameter settings for the learner, such as the kernel width in an RBF network. The most

useful parameter setting produces a hypothesis with the lowest VC-Dimension, as that facilitates fully automatic parameter tuning and does not require expensive cross-validation (Vapnik, 2013).

The formula used for the output of an SVM that is linear is u = $\vec{w}.\vec{x} - b$, where $\vec{w}$ is the normal vector to the hyperplane, and $\vec{x}$ is the input vector. For the linear case, the margin is the distance of the hyperplane to the nearest examples, whether positive or negative, and increasing the margin can be done through optimisation as follows: minimise $\frac{1}{2} \|\vec{w}\|^2$ subject to $y_i\ (\vec{w}.\vec{x} - b\ ) \geq 1$ , $\forall\ i$ where $x_i$ is the $_i$th training example and $y_i$ is the correct output of the SVM for the $_i$th training example.

### 2.3.2. Neural networks

Neural network text classifier refers to a network of units, with the input units representing terms; the output units representing categories of interest, and the weights around the edges that link units representing dependent relations. In order to classify a test document $d_j$, it is necessary to load its term weights $w_{kj}$ into the input units. These units are activated by propagating them forward through the network, with the value of the output units determining the decisions on categorisation. Training NNs is often done using backpropagation, which involves the weights of a training document being loaded into the input units, and in case of a misclassification, the error is "backpropagated" to prevent changing the network parameters and prevent or reduce the error. The perceptron is a linear classifier, and it is the simplest type of NN classifier (Dagan et al., 1997). Alternative types of linear NN classifiers are available for logistic regression, and these have been suggested by (Schütze et al., 1995) and (Wiener et al., 1995) who found they are effective and have a very

good yield. Alternatively, non-linear NN (Lam and Lee, 1999, Weigend et al., 1999) is a network that has at least one additional layer of units, where the training condition typically represents higher-order interactions between terms which the network cannot learn. Comparative experiments linking non-linear NNs to linear NNs have shown no improvements to non-linear NNs (Schütze et al., 1995), or negligible improvements compared to linear NNs (Wiener et al., 1995).

### 2.3.3. Decision tree classifiers

Probabilistic methods are quantitative and therefore are not that effective due to being difficult to interpret manually, although a class of algorithms that avoid this problem are symbolic, or non-numeric algorithms. Inductive rule learners and decision tree learners are the most widely used examples of non-numeric algorithms. A decision tree (DT) text classifier has internal nodes that are labelled, with branches moving off from them known as tests that check the weight of the term in the test document, and leaves are labelled according to categories (Mitchell, 1997). This type of classifier categorises test documents $d_j$ by recursively testing the weights that the terms label and checking their internal nodes in vector $d_j$, until a leaf node is attained, with this node's label then assigned to $d_j$. These types of classifiers usually involve binary document representations, and are therefore made up of binary trees. Several standard packages can be used for DT learning, and the majority of DT approaches to TC have used such a package, for example ID3 (Fuhr et al., 1991), C4.5 (Joachims, 1998) (Cohen and Hirsh, Cohen and Singer, Joachims, Lewis and Catlett), and C5 (Li and Jain). With regard to TC efforts based on experimental DT packages, Dumais et al. (Dumais et al.); Lewis and Ringuette (Lewis and Ringuette), and Weiss et al. (Weiss et al.) have explored

these. One method that may be used to learn a DT for category $c_i$ involves the divide and conquer strategy, which requires: (i) checking that all the training examples have been given the same label (either $c_i$ or $\overrightarrow{ci}$); otherwise, select a term $t_k$, and separate classes of documents with the same value for $t_k$ by putting each class into a separate subtree. The process is repeated again for the subtrees until every leaf generated has training examples assigned to the same category $c_i$, and this then forms the label of a certain leaf on the tree. An essential key step is to choose the term $t_k$ on which to carry out the split, and this is usually done using information gain or entropy criterion; however, a "fully grown" tree could be susceptible to overfitting due to some branches being too specific for the training data. Therefore, the majority of DT learning methods incorporate one method for growing the tree and one for removing overly specific branches (pruning). There are many variations of this basic schema used in DT learning (Mitchell, 1997). DT text classifiers can be the main classification tool (Fuhr et al., Lewis and Catlett, Lewis and Ringuette), or used either as baseline classifiers (Cohen and Singer, Joachims), or as members of a classifier committee (Li and Jain, Allwein et al., Weiss et al.).

### 2.3.4. Naïve Bayes Classifier

A simple Bayesian classification algorithm is the Naive Bayes classifier (Lewis and Ringuette, 1994), which has been shown to be useful for text categorisation, and to solve the text categorisation problem, one document d ∈ D corresponds to one data instance, with D referring to the training document set. Document d can be represented as a selection (bag) of words, with each word w ∈ d coming from a set W of feature words, and every document

d is linked to a class label c ∈ C, with C denoting the class label set. Naive Bayes classifiers are used to estimate the conditional probability P(c|d) that a document d belongs to a class c.

$$P(c|d) = P(c).P(d|c)$$

Furthermore, a major assumption regarding Naive Bayes classifiers is that the words in the documents are independent of the class value.

$$P(c|d) = P(c) \prod_{w \in d} P(w|c)$$

A popular way of estimating P(w|c) is by using Laplacian smoothing:

$$P(w|c) = \frac{1 + n(w, c)}{|w| + n(c)}$$

where the number of the word positions is n(w, c), and these are occupied by w in all of the training examples with a class value of c. n(c) is the number of word positions from class value is c, and |W| is the total number of distinct words contained in the training set.

A number of extensions to the Naive Bayes classifiers have been suggested, for example Nigam et al. (2000) combined the Expectation-Maximization (EM) (Dempster, Laird, & Rubin 1977) and the Naive Bayes classifiers for learning from a semi-supervised algorithm for both labelled and unlabelled documents. The EM algorithm maximises the probability of both labelled and unlabelled data. In addition, the algorithm used by Nigam et al. (2000) was applied by Rigutini et al. (2005) to deal with the cross-lingual text categorisation issues, and

they suggest a heuristic approach using Spy-EM for learning how to cope with training and test data that has non-overlapping class labels.

### 2.3.5. K-Nearest Neighbour Classifier

The text classification K-Nearest Neighbour (KNN) is a straightforward approach that involves providing a test document d, for which the system locates its K-nearest neighbours from the training documents, with the classes of the K-nearest neighbours used to weight class the candidates (Tan, 2005). The similarity score for every neighbour document close to the test document affects the weight of the classes of the neighbour document, and where a number of K-nearest neighbours share the same class, the individual neighbour weights of that class are taken and added together, with the result used for the likely score for that class with regard to the test document. In this way, sorting the candidate classes' scores produces a ranked list for use with the test document. The KNN classification decision rule in can be written as:

$$\text{score}(d, c_i) = \sum_{d_j \in KNN(d)} Sim(d, d_j)\ \delta\ (d_j, c_i)$$

The KNN(d) score that is achieved reveals the K-nearest neighbours of document d. $\delta(d_j, c_i)$ stands for the classification of document $d_j$ concerning class $c_i$, which is:

$$\delta(d_j, c_i) = \begin{cases} 1, & d_j \in c_i \\ 0, & d_j \notin c_i \end{cases}$$

In addition, test document d needs to be assigned to the class with the highest weighted sum result.

## 2.4. Performance Measures

### 2.4.1. Accuracy, Recall, Precision and F1 measures

the evaluation of text classifiers is typically conducted experimentally, rather than analytically. Precision (P) and recall (R) have been used regularly to measure the performance of information retrieval and information extraction systems. Precision deals with substitution and insertion errors while recall deals with substitution and deletion errors. Because of the community's desire to have a single measure of performance that deals with all three types of errors simultaneously – substitutions, deletions, and insertions – a single figure of merit, the F-measure, has been defined as a weighted combination of P and R (Sebastiani, 2002).

***Recall*** refers to the proportion of Real Positive cases which are Predicted Positive. However, it is often not considered for Information Retrieval due to the assumption that there will be a high number of relevant, so it is not important which subset is found, and it is not possible to know whether documents that are not returned are relevant or not. Thus, Recall is often ignored or averaged away in machine learning and computational linguistics, with approaches concentrating on confidence levels and how reliable the rule or classifier is. Even so, within a medical perspective, Recall is a primary approach, because the goal is to identify all Real Positive cases. Such measures and various combinations of them, concentrate on positive examples and predictions only, yet they can also capture information on the rates and types of errors that occur, but despite that, neither reveals information on how successfully the model copes with negative cases. ***Recall*** only deals with the condition positive column and ***Precision*** only to the predicted condition positive row, and these do not consider the number of True Negatives. The F1-measure is good at referencing the True Positives to the Arithmetic Mean of Predicted Positives and Real Positives, as it is a constructed rate that is normalised

to an idealised value, and when stated in this format, it is known in the field of statistics as a Proportion of Specific Agreement, because it is applied to a specifically to the Positive Class (Sebastiani, 2002).

It is possible to calculate accuracy, recall, F1-measures, and precision using the method set out below, and additional details may be added to clarify the measures applied using a confusion matrix:

1) Accuracy = (TP+TN) / (TP+FP+TN+FN)
2) Precision = TP / (TP+FP)
3) Recall = TP / (TP+FN)
4) F1 = 2× (Precision × Recall) / (Precision + Recall)

*Table 2-1 Confusion matrix*

| Confusion matrix | | Predicted Class | |
|---|---|---|---|
| | | Positive | Negative |
| Actual Class | Positive | True Positive (TP) | False Negative |
| | Negative | False Positive (FP) | True Negative |

### 2.4.2. McNemar's Test

One type of the $\chi^2$ test is McNemar's test (McNemar, 1947, Clark and Clark, 1999), which is a non-parametric test for analysing matched pairs of data. McNemar's Chi Square is the most powerful way to test model differences for supervised learning algorithms (Dietterich, 1998), and for McNemar's test, it is possible for two algorithms to have four potential outcomes arranged in a two-by-two contingency table as shown in Table 2-2.

*Table 2-2 Possible results of two algorithms*

| | Algorithm A failed | Algorithm A succeeded |
|---|---|---|
| Algorithm B failed | $N_{ff}$ | $N_{sf}$ |
| Algorithm B succeeded | $N_{fs}$ | $N_{ss}$ |

$N_{ff}$ refers to the number of instances when both algorithms failed, and $N_{ss}$ points out the success of both algorithms. However, these two scenarios do not provide enough detail on the algorithm's performance due to not providing information on how the performances may vary. On the other hand, the other two parameters ($N_{fs}$ and $N_{sf}$) reveal whether an algorithm has failed, and if another succeeded, which highlights any discrepancies in performance. To quantify the differences, McNemar's test uses the z score, as shown in the equation below:

$$Z = \frac{(|N_{sf} - N_{fs}| - 1)^2}{(N_{sf} + N_{fs})}$$

To interpret z scores: When z = 0, the two algorithms have similar performance, and as this value diverges from 0 in a positive direction, it shows that their performance is significantly different, and scores may be used to reveal confidence levels.

## 2.5. Learn to Search (L2S)

One group of algorithms that can solve structured prediction problems is L2S, and these algorithms have been shown to be very effective for solving problems with NLP such as part-of-speech tagging; named entity recognition (Daumé III et al., 2014); co-reference resolution (Ma et al., 2014), and dependency parsing (He et al., 2014).To summarise, L2S can be used

for structured predictions by decomposing the structured output's production with regard to an explicit search space, such as space or actions, and it can be used for learning hypotheses that have control over a policy involving action in the search space.

L2S focuses on "what is the best next action ($y_i$) to take" in a search space according to the current state. First there needs to be an initial policy on a trajectory (which is a rollin policy), before a one-step deviation is taken, followed by finishing the trajectory using another policy, which is the rollout policy. A number of variations of L2S have been defined according to what type of policies it uses throughout rollin and rollout; for example DAGGER, which uses rollin = learned policy and rollout = reference policy, or SEARN, which uses a rollin = rollout = stochastic mixture of reference and learned policy, as well as LOLS, which uses rollin=learned policy and a rollout=stochastic mixture of both reference and learned policy (Rao et al., 2016).

Learning to search algorithms usually operate based on the consideration of a search space, such as the one presented in Figure 2-2. The learning algorithm first uses a roll-in policy $\pi^{in}$ for some changes into state R, before assessing all of the possible actions available in state R. Next, it uses a roll-out policy $\pi^{out}$ until the sequence has finished. For fully supervised cases, the learning algorithm is able to compute the loss for all potential outputs, and this loss can be used to facilitate learning at state R, as the learner is encouraged to take the action at the lowest cost, with the learned policy updated from $\hat{\pi}_i$ to $\hat{\pi}_{i+1}$.

A summary of L2S algorithm is shown in Figure 2-3, which started with a pre-trained reference policy with the goal of looking to improve it through feedback. For all of the examples, an exploration algorithm decides if it is necessary to explore or exploit, and exploit is chosen, a random learned policy is used for making predictions, with no updates carried out. On the other hand, if the choice is to explore, it carried out a roll-in a single deviation at time t, in

accordance with the exploration policy, followed by a roll-out. Once completed, a loss occurs for the whole trajectory, and a cost estimator is used to estimate the cost from the action not being taken. From this, a complete cost vector is formed, with the underlying policy updated based on that cost vector, and then, finally, the cost estimator is updated (Sharaf et al.,2017).

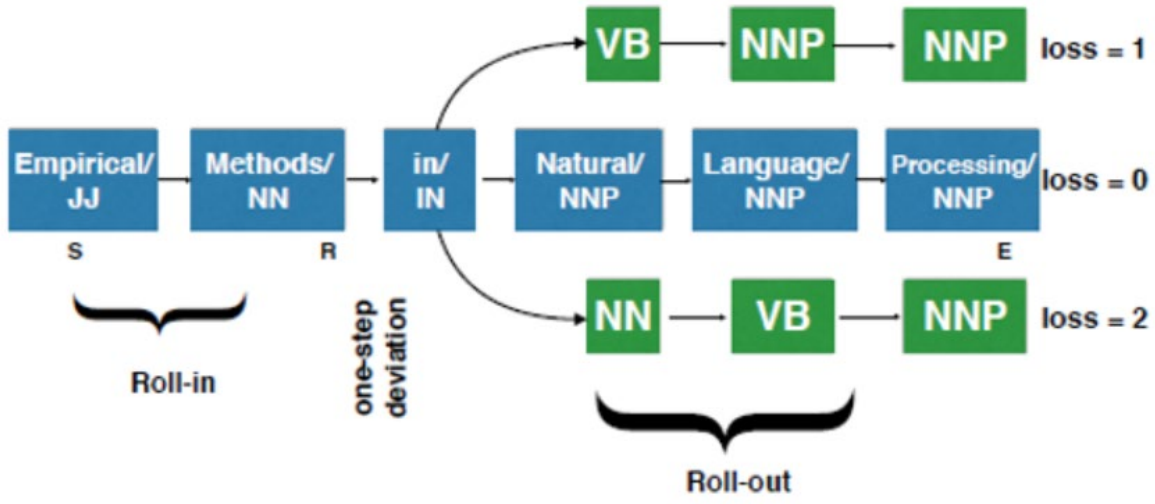


*Figure 2-2 A search space for part of speech tagging, explored by a policy that chooses to "explore" at state R (Sharaf et al.,2017).*

The easiest way to carry out cost estimation is importance sampling (Horvitz et al.,1952), whereby the third action is explored according to probability $\mathcal{p}_3$, and cost $\hat{c}_3$ is noted, with the cost vector for all actions set at $\langle 0,0,\hat{c}_3\ /\mathcal{p}_3\ ,\ 0,\ldots,0\rangle$. In this way, an unbiased estimate of the true cost vector can be obtained as by expecting every possible action, the cost vector is equal to $\langle \hat{c}_1, \hat{c}_2, \ldots, \hat{c}_k \rangle$. (Sharaf et al.,2017).

**Input:** examples $\{x_i\}_{i=1}^{N}$, reference policy $\pi^{\text{ref}}$, exploration algorithm $explorer$, and rollout-parameter $\beta \geq 0$

$\pi_0 \leftarrow$ initial policy

$\mathcal{I} \leftarrow \emptyset$

$\rho \leftarrow$ initial cost estimator

**for** *each* $x_i$ *in training examples* **do**

  **if** *explorer* chooses not to explore **then**

    $\pi \leftarrow$ *Unif*$(\mathcal{I})$ // pick policy

    $y_i \leftarrow$ predict using $\pi$

    $c \leftarrow$ bandit loss of $y_i$

  **else**

    $t \leftarrow$ *Unif*$(0, T-1)$ // deviation time

    $\tau \leftarrow$ roll-in with $\hat{\pi}_i$ for $t$ rounds

    $s_t \leftarrow$ final state in $\tau$

    $a_t =$ *explorer*$(s_t)$ // deviation action

    $\pi^{\text{out}} \leftarrow \pi^{\text{ref}}$ with prob $\beta$, else $\hat{\pi}_i$

    $y_i \leftarrow$ roll-out with $\pi^{\text{out}}$ from $\tau + a_t$

    $c \leftarrow$ bandit loss of $y_i$

    $\hat{c} \leftarrow$ *est_cost*$(s_t, \tau, \rho, A(s_t), a_t, c)$

    $\hat{\pi}_{i+1} \leftarrow$ *update*$(\hat{\pi}_i, (\Phi(x_i, s_t), \hat{c}))$

    $\mathcal{I} \leftarrow \mathcal{I} \cup \{\hat{\pi}_{i+1}\}$

    update cost estimator $\rho$

  **end**

**end**

*Figure 2-3 Learning to Search algorithm (Sharaf et al.,2017)*

# Chapter 3: Literature Review

## 3.1. Introduction

Detecting an event from the social media networks, especially Twitter, has attracted considerable research interest. Research efforts have focused on location inferencing, named entity recognition (NER), and new events detection. This chapter first presents work related to text classification (TC), and then an application of a SLR to review Arabic text classification. Secondly, the chapter provides an investigation of several existing models and techniques for the task of event detection using social media, as well as identifying the shortcomings of current approaches, paying attention to new event detection. Thirdly, a review of the techniques on location inference, especially using Arabic text, and some research on NER on Arabic Language, is presented. Finally, related works concerning Arabic NER on social media networks are described.

## 3.2. Text classification

Text classification or categorisation is the process of assigning a text document to one or more predefined classes based on its content. TC falls at the crossroads of Machine learning (ML) and Information retrieval (IR). There has been tremendous interest in this research area due to the large amount of textual data posted and widely shared on the World Wide Web (Chakravarty, 2010). TC has been used in different applications including topic identification, spam filtering and sentiment analysis. While some data can be described as being static, such as PDF files, HTML pages can be updated more frequently, while tweets are retrievable in real-time. In general, text classifiers can be categorised into two models: generative and discriminative. For instance Naïve Bayes (NB) is an example of a generative model that will first try to estimate the parameters from p(x | y) and p(y) from the training data, and then

calculate p(y | x) using Bayes theorem; where p(x | y) stands for a conditional probability of x given y is true. It is called “generative” since it can generate new samples by sampling from the learned joint distribution p(x, y). In contrast, a discriminative model estimates the parameters of p(x | y) directly from the training data without assuming anything about the input distribution p(x); such models include Support Vector Machines (SVM), Neural Networks and Decision Trees (Jordan, 2002, Raina et al., 2003). SVM is considered a non-probabilistic binary linear classifier, which can be used for either classification or regression. For a given set of training samples, the SVM model provides a representation of these samples as mapped points in space, isolated by a gap to distinguish the different categories. Likewise, Decision Trees can be used as a predictive model. Their structure includes leaves to represent classes (target values) and branches to represent conjunctions of features. However, in complex classification tasks, trees could fail to generalise from the training data (overfitting) or correctly illustrate a concept.

Furthermore, these two approaches can be combined to create a hybrid model, known as Generative-Discriminative Pairs (CDP). This is a connection between a generative model and a discriminative model where one can be directly transformed to the other (Raina et al., 2003). Examples include the Discriminative Hidden Markov Model (D-HMM) (Xue, 2008) and the pair of Naive Bayes together with Logistic Regression, in which a model is trained by optimising a combination of the generative and discriminative log likelihood functions to classify text. CDP has many advantages for addressing practical challenges, and (Hospedales et al., 2013) developed a hybrid model that can switch between generative and discriminative algorithms systematically as a subtask of the learning process; this has allowed them to achieve better results while discovering rare categories in a given dataset.

While discriminative classifiers often outperform their generative counterparts in accuracy, generative models have several advantages. It is assumed that they are easier for classifying data and can achieve better accuracy when the training data is limited (Raina et al., 2003). However, a generative approach produces a probability density model over all variables in a system and manipulates it to compute classification. While the overall design of generative models has the advantage of being more complete by definition, it can be wasteful and non-robust (Jebara, 2001). A discriminative approach makes no clear attempt to model the underlying distributions of the features in a system, and is only interested in optimising mapping from the inputs to the required class. As such, learning (not modelling) is the focus of discriminative approaches, which often lack flexible modelling; its techniques could feel like black-boxes, where the relationships between variables are not as explicit as in generative models (Jebara, 2001).

### 3.3. Arabic Text classification (Systematic Literature Review)

Although TC remains an active research area with novel techniques designed and tested on English scripts (Singla et al., 2014), there seems to have been very little work done on Arabic texts. Therefore, with the absence of a Systematic Literature Review (SLR) based on a comprehensive search protocol and quality assessment, it is not possible to determine the exact research gap for Arabic text, and this has become one of the objectives of this study and one of the key contributions of this thesis. For instance, it is important to point out better performing classifiers, and text pre-processing and Dimensionality Reduction Techniques (DRT) (Tu and Xu, 2012) have been proven to be more effective for Arabic.

Recent publications in this growing area of research include (Al-Anzi and AbuZeina, 2016) work on enhancing classifier's performance with Arabic text using cosine similarity and

latent semantic indexing; the effect of pre-processing on Arabic document categorisation by Ayedh et al. (Ayedh et al., 2016) and others (Kanan and Fox, 2016, Mohammad et al., 2016, Abainia et al., 2015).

The remainder of this section covers the methodology, in subsection 3.3.1, which also discusses the research questions in this study, the protocol used and, finally, the data extraction strategy. Section 3.3.2 contains SLR results analysis and the discussion of key findings from the primary studies included. Finally, colloquial Arabic Text classification is presented in Section 3.3.3.

### 3.3.1 Methodology

The research method is based on the SLR guidelines for the discipline of computer engineering as proposed by Kitchenham and Charter (Kitchenham and Charters, 2007). The key phases followed are demonstrated in Figure 3-1. Further reflection on each are shared within the consequent sections. In general, the problem statement, research questions and fundamental aspects of the review protocol are identified as part of the planning phase. To mitigate subjectivity, each of the phases was initiated only after the full evaluation and approval of the previous one. The search strategy consisted of the study selection criteria, procedure, unified search string and study quality assessment. The third phase is mainly concerned with the development of the Data Extraction strategy, and the final phase of the systematic review involved data synthesis and critical analysis.


*Figure 3-1 Main stages followed in this SLR.*

#### *3.3.1.1 SLR Research questions*

The main aim of this study has been achieved through answering the research questions defined and discussed below:

**SLR RQ1.** What TC models have been applied to Arabic text and supported by empirical evidence to estimate their accuracy? And which models perform better with Arabic text?

**SLR RQ2.** What characteristics can be identified to describe corpuses, techniques and algorithms that can affect the accuracy of these TC models?

The term 'models' used in the questions above could be used interchangeably with 'methods' and 'techniques'. Answering RQ1 will help to conclude a list of all relevant TC methods within the scope and requirements of this study, while RQ2 investigates their key characteristics. RQ1 helps to assess the accuracy of their implementation, and therefore reliability, in a real-life application. Both RQ1 and RQ2 will help to identify the gap in the current literature and suggest areas for further investigation. To frame these research questions effectively, PICOC criteria (Population, Intervention, Comparison, Outcome, and Context) (Kitchenham and Charters, 2007, Higgins and Green, 2008) have been applied from the viewpoint of software engineering as follows:

***Population*** Text Classification Models

***Intervention*** Generative and Hybrid models

***Comparison*** Discriminative models

***Outcomes*** Accuracy of the models analysed

***Context*** Academic research

#### *3.3.1.2 Data sources and search strategy*

Pioneer database sources for software engineering research publications have been used, as shown in Table 3-1. This study began in January 2015 and therefore publications up to that date have been considered. Searching keywords were defined to include the following key terms and synonyms constructed with logical operators to return the best possible search outcome:

*('Arabic text' OR 'Arabic script') AND ('classification' OR 'Classifier' OR 'categorization' OR 'categorisation').*

*Table 3-1 DATABASES*

| ***Database*** | ***URL*** | ***Database*** | ***URL*** |
|---|---|---|---|
| *IEEExplore* | http://ieeexplore.ieee.org | Academic Search Elite | https://www.ebscohost.com/ |
| *ACM Digital library* | http://dl.acm.org | DOAJ | https://doaj.org/ |
| *CiteSeerX library* | http://citeseerx.ist.psu.edu/index | Web of Knowledge | http://www.webofknowledge.com |
| *Science Direct* | http://www.sciencedirect.com | Scopus | http://www.scopus.com/ |
| *Springer* | http://link.springer.com/ | Google scholar | http://scholar.google.co.uk |

This search string was adapted to the built-in options of each database shown in Table 3-1 to filter and refine the results. Furthermore, the grey literature was considered in the search

strategy, together with a snowballing approach (reference of references) whereby any paper collected by the search criteria can manually lead to another reference from within its bibliography.

#### *3.3.1.3 Study selection criteria*

In this step, rigorous inclusion and exclusion criteria were applied to ensure valuable and relevant information in response to the research questions defined. These criteria were enforced after reading the title, abstract and then the full text of the articles, as demonstrated in the study selection procedure shown in (3.3.1.43.3.1.4). For instance, (Al-Shawakfa et al., 2010) has been excluded because it does not report the method's accuracy and (Mamoun and Ahmed, 2014), as that is not a primary study.

Inclusion criteria:

- Must be a primary study reporting on TC models in the area of software engineering /data mining.
- Must address the accuracy of the TC model/method.
- Must include analysis and empirical evidence.

Exclusion criteria:

- Publication is not peer reviewed.
- Arabic is not the language used to test the TC model.

#### *3.3.1.4 Study selection procedure*

The selection of the primary studies was examined by all authors. Four different phases show how the selection procedure was implemented, as illustrated in Figure 3-2:

***Phase 0 – Keywords-based filtering:***

In this phase, the search string was applied to the ten scholarly databases shown in Table3-1. This yielded a total of 1464 articles, which were included in the next phase.

***Phase 1 –Title, indexing keywords and abstract-based filtering:***

In this phase, the titles were examined against the inclusion and exclusion criteria. Articles deemed to be of any relevance were directly included in the next phase. In conclusion, 863 articles were discarded and 365 articles were included.

***Phase 2 – Full text-based filtering:***

This is the final stage in which the reviewers discussed and resolved any disagreements regarding the relevance of the articles to the current study. A total of 192 articles were identified as being duplicates downloaded from different databases, and were therefore discarded. Upon reconsideration of the inclusion and exclusion criteria, 125 articles were excluded for different reasons; for instance, (Alaa, 2008) was not peer reviewed, (Yahia, 2011) did not include an empirical study and (Ben Othmane Zribi et al., 2010) did not satisfy a number of the quality assessment criteria shown in (3.3.1.5). The final set for the primary study included a total of 48 remaining articles.

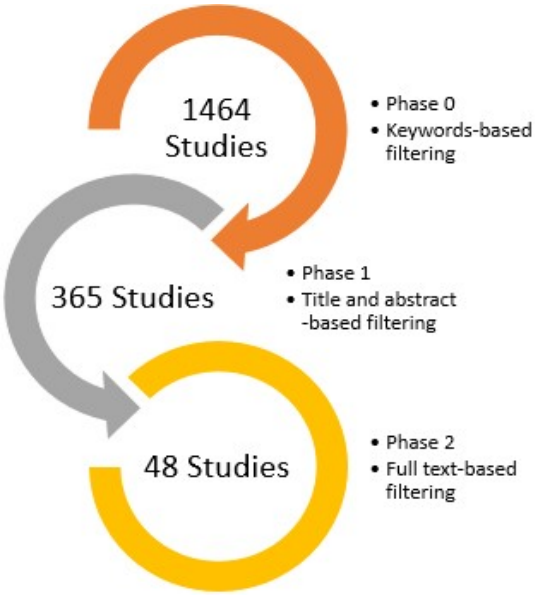


*Figure 3-2 The number of primary studies included in each phase of the study selection procedure*

#### *3.3.1.5 Study quality assessment*

The papers included had to satisfy the quality assessment designed as a measure to determine whether a given paper is suitable for addressing the research questions. The following checklist had to be met with affirmative answers:

- Has the number of training and testing data been identified?
- Are the pre-processing techniques used in the study clearly described and their selection justified?
- Are the classifiers used in the study clearly described?
- Is there comparison with other approaches?
- Are the performance measures fully defined?

#### *3.3.1.6 Data extraction strategy*

The data extracted from the studies was tabulated and comprises the following characteristics: year of publication, number of learning and testing documents, feature selection approaches, classification algorithm, and accuracy.

### 3.3.2 Results analysis and discussion

#### *3.3.2.1 Primary studies*

There were a total of 48 studies included in the form of journal articles and conference proceedings published between 2006 and 2014. The results analysis shows that most articles were published within the last five years, which is an early indication that Arabic TC is an active research area and has started to evolve very recently. More details are provided in Table 3-2.

*Table 3-2 THE DISTRIBUTION OF PRIMARY STUDIES BY PUBLICATION TYPE AND PUBLICATION YEAR*

| Source | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| journal | 1 | 2 | 2 | 3 | 2 | 4 | 3 | 1 | 7 | 25 |
| conference | 1 | 0 | 2 | 2 | 5 | 3 | 4 | 6 | 0 | 23 |
| % | %4 | %4 | %7 | %10 | %15 | %15 | %15 | %15 | %15 | %100 |

#### *3.3.2.2 Key focus areas*

This SLR found that primary studies can be classified by their main focus area into four domains: TC algorithms, Features Selection (FS), Stemming Techniques (ST) and Term Weighting (TW). The majority of work has been on TC algorithms, as shown in Table 3-3. Each of these main areas are discussed in detail in the following sections.

*Table 3-3 KEY FOCUS AREA FOR INCLUDED PAPERS*

| Focus area | % | Studies |
|---|---|---|
| TC algorithms | 56 | (Khorsheed and Al-Thubaity, 2013, Belkebir and Guessoum, 2013, AlSaleem, 2013, Kadhim and Omar, 2012, Al-Thubaity et al., 2008, Altheneyan and Menai, 2014, Hmeidi et al., 2008, Majed et al., 2011, Thabtah et al., 2011, Al-Jaloud et al., 2012, Alwedyan et al., 2011, Al-Shargabi et al., 2011, Hadi et al., 2010, El-Halees, 2007, Duwairi, 2007, Abbas et al., 2010, Al-Harbi et al., 2008, Alsaleem, 2011, Al-Shalabi et al., 2006, Kanaan et al., 2009, Harrag and Al-Qawasmah, 2010, Al-Thwaib, 2014a, Hmeidi et al., 2014, Duwairi and El-Orfali, 2014, Mahafdah et al., 2014, Al-diabat, 2012, Zrigui et al., 2012) |
| Features Selection (FS) | 25 | (Al-Thubaity et al., 2013, Al Zaghoul and Al-Dhaheri, 2013, Elberrichi and Abidi, 2012, Harrag and El-Qawasmah, 2009, Mesleh, 2011, Harrag et al., 2010, Al-Thubaity et al., 2012, Saad and Ashour, 2010, Al-Thwaib, 2014b, Duwairi, 2013, Raheel and Dichy, 2010, Al-Shalabi and Obeidat, 2008) |
| Stemming Technique (ST) | 13 | (Hadni et al., 2012, Duwairi et al., 2009, Al-Shammari, 2010, Omer and Ma, 2010, Al-Kabi et al., 2013, Nehar et al., 2013) |
| Term Weighting (TW) | 6 | (Duwairi, 2006, Khreisat, 2009, Ababneh et al., 2014) |

#### *3.3.2.3 Data collection (Corpus)*

Collecting data to create a suitable dataset is the first step in text classification studies. There are several free benchmarking datasets for English used for TC purposes: the 20 Newsgroup contains around 20,000 texts distributed almost evenly into 20 classes; Reuters-21578 contains 21,578 texts belonging to 17 classes; and RCV1 (Reuters Corpus Volume 1), contains 806,791 texts classified into four main classes (Khorsheed and Al-Thubaity, 2013). Unfortunately, the case is different for Arabic. There seems to be no free benchmarking

dataset identified from the studies included for the Arabic TC. For most research, authors have collected data to build their very own datasets, mostly from online formal websites and news articles. Table 3-4 describes the datasets used in each study. It also shows the language model selected, and whether it is classical Arabic (also known as Qur'anic Arabic), which could also include old poems and religious scripts; modern Arabic currently used in formal press and government communications; colloquial Arabic as in informal local dialects; or a mixture of these. It has also been noted that some papers do not seem to describe their datasets enough, which makes it difficult to classify them. Such works usually do not publish their data for other researchers to utilise. Consequently, confidence in the results derived from such experimental studies is not satisfactory enough. The performance of the data mining approaches adopted is biased towards such data sets and could be ambiguous.

The results show that most work has been conducted in the modern language with a single study (Duwairi and El-Orfali, 2014) covering informal (colloquial) writing; this is an interesting finding because it covers a huge and critical technology gap, as informal Arabic is what people use on social media, especially Twitter. Arabic dialects vary from one Arab country to another and could also vary slightly between cities and towns.

With regards to the size of datasets, they ranged from 119 documents divided into three classes (Saad and Ashour, 2010) to 17,652 documents divided into six classes (Al-Thubaity et al., 2008). The vast majority of studies have measured the size according to the number of documents rather than word count. This detail gives an indication of the size, but it remains a challenge to obtain an accurate statistical comparison between the different datasets used.

*Table 3-4 SOURCES FOR BUILDING DATASETS*

| Models | Corpus | Studies |
|---|---|---|
| Classic | Quran | (Mahafdah et al., 2014) |
| | Religious scripts | (Harrag and El-Qawasmah, 2009, Harrag et al., 2010, Harrag and Al-Qawasmah, 2010, Hadi et al., 2010) |

| | | |
|---|---|---|
| | Old books | (Altheneyan and Menai, 2014) |
| Modern | Websites | (Belkebir and Guessoum, 2013, Al Zaghoul and Al-Dhaheri, 2013, Kadhim and Omar, 2012, Hadni et al., 2012, Duwairi et al., 2009, Khreisat, 2009, Omer and Ma, 2010, Al-Kabi et al., 2013, Duwairi, 2013, Raheel and Dichy, 2010) |
| | News articles | (Al-Thubaity et al., 2013, AlSaleem, 2013, Elberrichi and Abidi, 2012, Duwairi, 2006, Hmeidi et al., 2008, Al-Shammari, 2010, El-Halees, 2007, Duwairi, 2007, Alsaleem, 2011, Al-Shalabi et al., 2006, Kanaan et al., 2009, Saad and Ashour, 2010, Al-Thwaib, 2014b, Al-Thwaib, 2014a, Hmeidi et al., 2014, Nehar et al., 2013, Al-diabat, 2012, Zrigui et al., 2012, Ababneh et al., 2014) |
| Colloquial | User Reviews | (Duwairi and El-Orfali, 2014) |
| Hybrid | | (Khorsheed and Al-Thubaity, 2013, Al-Thubaity et al., 2008, Thabtah et al., 2011, Al-Harbi et al., 2008) |
| Unknown | | (Mesleh, 2011, Majed et al., 2011, Al-Thubaity et al., 2012, Al-Jaloud et al., 2012, Alwedyan et al., 2011, Al-Shargabi et al., 2011, Abbas et al., 2010, Al-Shalabi and Obeidat, 2008) |

#### *3.3.2.4 Text pre-processing and dimensionality reduction techniques*

Pre-processing is a trial carried out to improve text classification by removing worthless data. It may include the removal of numbers, punctuation (e.g. hyphens) and stop-words (e.g. prepositions and pronouns). Due to its writing style, Arabic requires careful strategies at this stage to normalise writing forms and remove diacritics.

A number of dimensionality reduction techniques have also been used to reduce the number of terms included for analysis (classification); high dimensionality data do not satisfy the requirements of TC methods to produce reasonably accurate outcomes, and are therefore considered problematic (Sebastiani, 2002). The studies included have identified the use of two reduction techniques, namely: Stemming and Feature Selection.

#### 3.3.2.4.1 Stemming

Stemming is a technique used to reduce the high dimensionality of the feature space in text classification. Several Stemming approaches exist for the Arabic language, and each produces a different set of roots. These are identified in Table 3-5 and discussed in further detail below.

**Root-based stemming (Lexical)** is based on removing all attached prefixes and suffixes in an attempt to extract the root of a given Arabic surface word. An example of this approach is the Khoja stemmer (Khoja, 2001). Its core-function works by mapping words into their root patterns. Root patterns in Arabic are three, four, five, or six-letter patterns. More than 80% of Arabic words can be mapped into a three-letter root pattern, and reducing a word to its root pattern could decrease the number of words from hundreds of thousands to as little as 4,749, as in (Duwairi, 2006).

*Table 3-5 STEMMER TECHNIQUES USED*

| Stemmer | Studies |
|---|---|
| Root-based stemming | (Duwairi, 2006, Thabtah et al., 2011, Al-Shammari, 2010, Al-Jaloud et al., 2012, Duwairi, 2007, Saad and Ashour, 2010, Duwairi and El-Orfali, 2014) |
| Light stemming | (Belkebir and Guessoum, 2013, Kadhim and Omar, 2012, Duwairi et al., 2009, Harrag and El-Qawasmah, 2009, Omer and Ma, 2010, Al-Shalabi et al., 2006, Harrag and Al-Qawasmah, 2010, Duwairi, 2013, Hmeidi et al., 2014) |
| Statistical stemmer | (Al-Shalabi and Obeidat, 2008, Raheel and Dichy, 2010, Mahafdah et al., 2014) |
| Hybrid | (Hadni et al., 2012) |

***Light Stemming*** does not attempt to give the linguistic root pattern of the word; instead, its main focus is on removing the most frequent suffixes and prefixes. There are different types of Light Stemming, and many studies have considered this approach (Table 3-5). The literature, in general, states the argument that light stemming allows remarkably good information retrieval (Larkey et al., 2007) and this is discussed in further detail.

***Statistical stemmer*** (character level N-Gram); N-Gram is a set of N consecutive characters extracted from a word. The main idea behind this approach is that similar words will have a high proportion of N-Gram in common. Typical values for n are 2 or 3, and these corresponding to the use of diagrams or trigrams, respectively. For example, when 3-grams is applied to the following string: “text classification”, the output is: “tex”, “ext”, “xt_”, “t_c”, “_cl “, “ cla”, “las”, “ass”, and so on (Hadni et al., 2012). Each of these strings will then be compared against the output of another string to measure and determine the level of similarity between the two.

***A hybrid approach*** was also tested, where a number of stemming techniques are used together in an attempt to improve the process. For example, (Hadni et al., 2012) proposed a hybrid method incorporating Khoja stemmer, light stemmer and N-Gram; the results were promising, with an improvement in the overall accuracy. Likewise, (Duwairi, 2006) used root extraction by assigning weights and ranks to the letters that constitute a word. However, they mention that roots are semantically weak in the meaning as several words can be mapped to the same root.

Nonetheless, in some cases, stemming techniques could decrease the performance of the classifier used. Kanaan et al., (Kanaan et al., 2009) observed this behaviour when light stemming was used with the Rocchio and NB algorithms. Likewise, Al-Kabi et al., (Al-Kabi et al., 2013) conducted an experiment and concluded that Khoja stemmer did not improve the classification accuracy for NB, SVM (SOM) and Decision Tree (J48).

##### 3.3.2.4.2 Feature selection

Some reduction methods utilise features (terms) selection to reduce dimensionality. These statistical techniques work at the term level, as such when 3-gram is utilised, and text

is split into chunks of three terms (words rather than characters). Table 3-6 demonstrates which FS techniques were used by each study.

*Table 3-6 FEATURE SELECTION TECHNIQUES*

| FS Techniques | Studies |
|---|---|
| Chi-square | (Al-Thubaity et al., 2013, Elberrichi and Abidi, 2012, Kadhim and Omar, 2012, Altheneyan and Menai, 2014, Hmeidi et al., 2014, Mesleh, 2011, Al-Thubaity et al., 2012, Al-Harbi et al., 2008, Raheel and Dichy, 2010) |
| Term Frequency | (Al-Shalabi et al., 2006, Al-Thwaib, 2014b) |
| Document Frequency | (Ababneh et al., 2014) |
| Information Gain | (Al-Thubaity et al., 2008) |
| N-gram | (Khreisat, 2009, Nehar et al., 2013, Duwairi and El-Orfali, 2014) |
| Hybrid | (Khorsheed and Al-Thubaity, 2013) |

Most studies have applied Chi-square (CHI), while there is a single study by (Khorsheed and Al-Thubaity, 2013) that attempted a hybrid approach in which the authors applied Document Frequency and Galavotti, Sebastiani, Simi (GSS).

#### *3.3.2.5 Feature representation (term weighting)*

TC algorithms require that text features are formatted before they can be interpreted by the specified classifier. This process is also referred to as term weighting because each term is entered together with a weight value. The papers included show that the most commonly used technique is the Term Frequency-Inverse Document Frequency (TF-IDF) (as in (Al-Shargabi et al., 2011, Abbas et al., 2010, Al-Shalabi and Obeidat, 2008, Al-Shalabi et al., 2006, Al-Kabi et al., 2013, Harrag and Al-Qawasmah, 2010, Duwairi, 2013, Raheel and Dichy, 2010, Duwairi and El-Orfali, 2014, Zrigui et al., 2012, Al-Thubaity et al., 2012, Harrag et al., 2010, Hmeidi et al., 2008, Harrag and El-Qawasmah, 2009, Al Zaghoul and Al-Dhaheri, 2013, Belkebir and Guessoum, 2013). It is a statistical method used to indicate the significance of a word within a given corpus. The utilisation of this technique is justified assuming the authors

wanted to weight terms while considering their significance across all documents, rather than a single one. Although in (Al-Thubaity et al., 2012), a simpler but more limited method has also been used to conclude a Boolean value of zero or one, and a term can be described as either important or not important. Whilst in TF-IDF, for a given term, a larger TF-IDF value indicates a more frequent word. As such, data can be represented as a matrix with n rows and m columns, wherein the rows correspond to the texts in the training data, and the columns correspond to the selected feature. The value of each cell in this matrix represents the weight of the feature in the text.

#### *3.3.2.6 Classification algorithms and accuracy*

Each study has used their very own corpus and different experimental conditions in terms of the training and testing procedure, pre-processing and DRT. Hence, it is not feasible to statistically compare accuracy values (cross studies). However, when analysing the outcomes of different studies, there is evidence that the Support Vector Machine (SVM) classifier (a discriminative model) outperforms other classifiers, with the exception of two studies reporting in favour of the C5.0 Decision Tree Algorithm, and one study on k-NN. This outcome is demonstrated in Table 3-7.

While all included studies have also reported the accuracy of their classifiers, Table 3-7 shows the accuracy values for each study, which have been reported in the following format: [study] (accuracy for the preeminent classifier – accuracy for the first method in comparison, accuracy for the second method …). The table also includes only those studies that attempted to conduct experiments on multiple algorithms within a controlled environment for comparison purposes.

The results show that generative models remain an option when the amount of training is relatively small and could therefore be faster; both algorithms that reportedly outperformed other models are discriminative (SVM and C5.0).

*Table 3-7 STUDIES INVESTIGATING ACCURACY*

| Preeminent Classifier | Compared with | Studies (and accuracy values) |
|---|---|---|
| SVM | NB | (Khorsheed and Al-Thubaity, 2013)(0.805-0.755), (Alwedyan et al., 2011)(0.778-0.74), (Hadi et al., 2010)(0.954-0.884), (Alsaleem, 2011)(0.778-0.74), (Raheel and Dichy, 2010)(0.9241-0.8949), (Al-Shammari, 2010) (0.8638-0.7741) |
| | k-NN | (Al-Thwaib, 2014a)(0.827-0.448), |
| | ANN | (Belkebir and Guessoum, 2013) (0.956- 0.94) |
| | NB, k-NN, ROCHIO | (Mesleh, 2011) (0.9141-0.8778, 0.7581, 0.7472) |
| | J48, NB | (Majed et al., 2011)(0.948-0.8942, 0.8507), (Al-Shargabi et al., 2011)(0.9608-0.9048, 0.856), (Al-Kabi et al., 2013)(0.896-0.753, 0.835) |
| | J48, NB, k-NN | (Hmeidi et al., 2014) (0.98-0.856, 0.967, 0.799) |
| | NB, k-NN | (Duwairi and El-Orfali, 2014) (0.611-0.585, 0.601), (Zrigui et al., 2012)(0.914-0.845, 0.727) |
| NB | ANN, k-NN | (AlSaleem, 2013) (0.85-0.81) |
| | k-NN | (Duwairi, 2007) (0.81-0.78), (Al-Thubaity et al., 2012)(0.8574-0.7995) |
| | k-NN, RACHIO | (Kanaan et al., 2009) (0.82-0.7871, 0.7882) |
| | SVM, k-NN | (Duwairi and El-Orfali, 2014) (0.857-0.824, 0.646) |
| Decision-tree (C5.0) | SVM, NB, ANN | (Al-Thubaity et al., 2008) (0.8443-0.761, 0.7566, 0.6378) |
| | SVM | (Al-Harbi et al., 2008) (0.7842-0.6865) |
| k-NN | SVM, NB | (Duwairi and El-Orfali, 2014) (0.666-0.598, 0.563) |

SVM is a supervised learning algorithm with an appropriate kernel. The algorithm can function competently, whether or not the data is linearly separable. It is widely used even with text of high-dimensionality. However, its disadvantages can be summarised as the algorithm's complexity, interpretability and memory requirements (Han et al., 2011). However, not all discriminative models performed better, and the K-Nearest Neighbor (k-NN) Classifier is an exemplar of this. It is discriminative because it models the conditional probability of data belonging to a given class. k-NN computes the similarities between a new sample and the

training samples previously stored in a dataset. The most K similar ones are then listed in descending order. Finally, the new sample takes the class label that belongs to the majority of these K neighbours (Al-Shalabi et al., 2006). It should therefore not be preferred for text categorisation (Han et al., 2011). Nonetheless, although C5.0 Decision Tree algorithm outperformed SVM, the latter outperformed another Decision Tree algorithm, J48, while many others remain untested in the literature.

As mentioned earlier, the remaining set of the studies included did not conduct a comparison between classifiers, but have instead investigated other factors. For instance, (Hadni et al., 2012) used NB with different stemmer techniques and found that a hybrid method gives more accurate results if compared to a root-based stemmer, light stemmer or n-gram (statistical stemming). Likewise, (Hmeidi et al., 2014) used SVM with different stemming techniques, however the study reports a very minor effect on accuracy.

More work needs to be done on Arabic text analysis as it can be applied to solve real-world problems, such as automating procedures, building intelligence, and mitigating cybercrime (Frommholz et al., 2016). Future work on TC techniques for Arabic text should ideally consider using a corpus that is available online for downloading, as this will enable comparative experiments by other researchers and conclude robust facts with regards to the accuracy and speed of the different algorithms and techniques available. Additionally, datasets should be described thoroughly, and sharing the word count to describe the size is the right approach rather than the number of documents collected.

Implementing a Hybrid Stemming and/or Feature Selection approach could improve the accuracy, as several studies suggest. The majority of papers report using root-based stemming, light stemming and Chi-square, therefore more research is needed to investigate the opportunities and threats for adopting hybrid Dimensionality Reduction Techniques for

Arabic text during both Stemming and Feature Selection. Not all discriminative algorithms outperform the accuracy of generative models. NB outperformed k-NN, however, both preeminent algorithms from the included studies were discriminative; SVM and C5.0. Additionally, no work has been found using the search protocol that compares with the hybrid model, Generative-Discriminative Pairs (CDP).

Furthermore, TF-IDF has been used in the vast majority of papers, but there is little discussion and justification for adopting this statistical method. It is critical that new research realises this limitation in the current literature, as the lack of detail was a key reason for excluding some papers according to the protocol, mainly because they have failed to describe their training datasets and report the accuracy of the algorithms utilised.

### 3.3.3 Colloquial Arabic Text Classification

Although Text Classification remains an active research area, with novel techniques designed and thoroughly tested on English scripts (Singla et al., 2014), there seems to have been very little work done on Arabic text compared to other languages, especially colloquial text, as shown in 3.3.2. However, as a growing area of research, recent publications include Huang's (Huang, 2015) work, who aimed to improve the classification of Arabic dialects through the utilisation of metadata such as records related to the geographical information of social media users. Another technique has combined three classifiers by computing the model scores of weakly supervised, strongly supervised, and semi-supervised classifiers. This combination has shown significant improvement by means of classification accuracy for both MSA and Colloquial Arabic test sets. An experiment by Kaati et al. (Kaati et al., 2015) was conducted to detect any tweet supporting an act of terrorism (a malicious agenda). The authors used data-dependent and data-independent features as part of the 'features

selection' process for Arabic and English tweets. Their results show that utilising AdaBoost (Adaptive Boosting) improves the performance for English datasets but not in the case of Arabic tweets. Furthermore, a framework combining classification and clustering techniques (Alsaedi et al., 2016) was proposed to enable the detection of real-world events from Arabic tweets. The authors have assessed their framework against (Becker et al., 2011) and (Pan and Mitra, 2011); the outcome of this comparison demonstrates the effectiveness of their proposed framework, as it outperformed other approaches in the Normalized Discounted Cumulative Gain (NDCG) and precision evaluation measures. In addition, the work of Al-Badarneh et al. (Al-Badarneh et al., 2016) investigated the impact of using different indexing techniques (full-word, stem, and root) when classifying Arabic text. It concludes that using 'full-word' or 'stem' outperforms 'root' when applied with the Naïve Bayes (NB) classifier.

Likewise, Shoukry and Rafea (Shoukry and Rafea, 2012) examined the sentiment classification of Arabic text at a sentence-level by comparing Support Vector Machine (SVM) and NB Classifiers before and after removing stop words. Their corpus of 1000 tweets comprised the usual classes of positive and negative tweets for training purposes. The conclusion of this study suggests that SVM has better results compared to NB. In (Al-Wehaibi and Khan, 2015), the performances of two classifiers were tested, namely NB and Decision Tree. The best classification model was obtained with the help of Decision Tree. The results demonstrate that light stemming is more suitable for use with colloquial Arabic text than other feature selection techniques. Another comparison study examining different classifiers is presented in (Hadi, 2015). A corpus of 3700 Arabic tweets was collected and partitioned into three different classes. The experiments conducted show that SVM outperformed k-NN, NB and Decision Tree in terms of accuracy. In (Brahimi et al., 2016), three classifiers SVM, NB, and k-NN were used to investigate the impact of feature selection on the performance of these

classifiers. The authors applied different feature selection techniques, including light stemming, root stemming, and character n-grams. The experiment utilised a dataset consisting of 1000 positives and 1000 negative tweets. The results conclude that 3-ngram and 4-ngram models without or combined with tokens (the word model) yield the best results. Regarding the classifiers' performance, SVM outperformed other classifiers when applying all the feature selection techniques included in their study.

### 3.4. Event Detection in Social Networks

The notion of an event refers to a unique incident or occurrence happening at some point in time (Allan et al., 1998). Event detection aims at finding, and following, events from conventional media sources or social media sources. As mentioned in 1.1.3 , Atefeh and Khreich (Atefeh and Khreich, 2015) classify event types as 'specified' or 'unspecified', detection tasks as 'Retrospective Event Detection (RED)' or 'New Event Detection (NED)', and detection methods as 'supervised' or 'unsupervised'. Since this work focuses on detecting flood events in real-time, the emphasis will therefore be on 'specified events' utilising 'supervised' methods to facilitate NED.

Specified events rely on partially or fully adding predefined filters based on information already known about the targeted event. These filters (or metadata) could include location, time, and keywords (Atefeh and Khreich, 2015); (Sakaki et al., 2010). While NED on Twitter involves the discovery of new Twitter streams in near real-time, and supervised learning methods relying on labelled training examples.

Manually labelling a large number of Twitter messages for training purposes is a labour-intensive and time-consuming task. It is also more feasible for specified events compared to unspecified events. Nevertheless, for experts to annotate a dataset of reasonable size, or for

processing resources to be utilised efficiently, good event descriptors and thoroughly tested filters have become a necessity to reduce the number of irrelevant messages.

### 3.4.1 New Event Detection

Several studies have been conducted to investigate the usability of NED. For instance, real-time detection of earthquakes and typhoons using Twitter users as 'social sensors' (Sakaki et al., 2010) was tested on Japanese tweets (considering the geographical location of interest). The research in question formulated NED as a classification problem and trained a SVM classifier on a manually labelled Twitter dataset comprising positive events (earthquakes and typhoons) and negative events (other events or non-events). Likewise, Popescu and Pennacchiotti (Popescu and Pennacchiotti, 2010) used a large set of linguistic, structural, sentiment, and controversy features from Twitter, and external features such as Web-news controversy for the detection of controversial events about celebrities.

Furthermore, Alsaedi and Burnap (Alsaedi and Burnap, 2015) proposed a framework based on Naïve Bayes (NB) classifier to detect 'disruptive' events from Arabic tweets in real time. Their system applied an incremental online clustering algorithm by calculating each tweet's similarity to existing clusters. In term of classifier accuracy, the authors report that the NB classifier outperformed SVM and Logistic Regression classifiers for both Arabic and English when using a combination of attributes (Named Entity Recognition (NER), Bigrams, Unigrams and part-of-speech (POS)). They note that, as a result of the limitation of POS and NER in the Arabic language, there was a drop in the recorded accuracy of all three classifiers, with Arabic compared to using English as input.

### 3.4.2 Event Location Inference on Twitter

The location inference of Twitter data has received considerable attention for many purposes, such as natural disasters (Sakaki et al., 2010), public health (Paul and Dredze, 2011) and political elections (Skoric et al., 2012). A number of studies have focused on social networking to infer location; (Li et al., 2012) proposed a model to infer profiling users' home locations by analysing users' social network and user tweets. Their data set included the social networks and tweets of 139,180 users; the average is 14.8 friends, 14.9 followers and 29 places per user. They state that their model improves the accuracy of two state-of-the-art methods (Backstrom et al., 2010) and in inferring user home location by 13% (Cheng et al., 2010).

(Davis Jr et al., 2011) focused on the social relationships between Twitter users to infer users' location; they executed their methods using random users mentioning of keywords related to "dengue fever" and 61,400 friends' users (following and follower). They found that from the last 10 tweets for each user, out of 61,400 users, 1.3% had at least one GPS location, 2.9% had at least one tweet that included geo IP information, and 37.3% had at least one tweet with location entity. The priority for location value was GPS, geo IP then location entity respectively. The experimental results show that out of 24,767 users with location information, their method could correctly predict around 40% of the locations for 91.4% of those users. They report that the reason for such low accuracy is that many users have no friends or only one friend.

Schulz et al. (Schulz et al., 2013) applied a multi-location indicator approach to infer tweet location. Their method is based on various weighted indicators. These indicators included using URL links, extracting location entities, user profile time zone and tweet geotag location. They used various location-based services such as Geonames gazetteer, a named entity

recognition service DBpedia Spotlight (DBpedia, 2017), and Foursquare service. They collected 500 tweets for testing purposes, and the results show that their method correctly estimated 92% of all tweets with a median of 29.66km of their actual locations.

Ryoo and Moon (Ryoo and Moon, 2014) proposed another approach to infer Korean Twitter users' profile locations using their textual content by applying a probabilistic generative model. Their model is based on inferring user location through an analysis of the spatial locality of words. The results show that their approach identified 60% of users within 10 km of their actual locations.

### 3.4.3 Arabic NER on Twitter

Named entity recognition (NER) is a natural language processing technique used to identify which snippets in a text mention entities in the real world. NER is difficult for user-generated content in general, and in the microblog domain specifically. (Derczynski et al., 2015) state that the conventional tools of NER perform poorly in the microblog domain compared to news texts and longer blog texts. This is due to the reduced amount of contextual information in short messages, less grammatical rules than longer posts, capitalisation errors, and the frequent use of emoticons, abbreviations and hashtags.

NER systems rely on various techniques including using gazetteer-based lookups such as the ANNIE system from GATE (Cunningham et al., 2002); DBpedia Spotlight (DBpedia, 2017); Lupedia (Ontotext, 2018); the use of a machine learning-based method to detect named entities such as Stanford NER system (Finkel et al., 2005); TextRazor (Txtrazor, 2018) and Alchemy API (IBM, 2018). NER over Twitter has only recently become an active research topic, and several approaches have been proposed for the English language, such as (Ritter et al., 2011) approach by applying tokenisation, POS tagging and topic models to find named entities; their T-NER system outperformed the Stanford NER system on a Twitter dataset.

The great challenge to NER systems when dealing with colloquial Arabic text is the ambiguity of words because of using undiacritized words, as Arabic words can have different meanings in different contexts when missing diacritics, which increases the complexity of NER Systems (Abdallah et al., 2012). While most existing Arabic texts are written in colloquial Arabic text, previous work has focused on NER from MSA text only and especially from news articles (Zayed and El-Beltagy, 2012); (Zirikly and Diab, 2014). (Abdallah et al., 2012) propose a simple integration between a rule-based system and a machine-learning classifier for Arabic named entity recognition. They re-implemented (Benajiba et al., 2008) work and then integrated it with a Decision Tree classifier. The experiment's results show that the integrated system achieved an 8-12% improvement over the rule based system.

(Zirikly and Diab, 2014) developed a NER system for colloquial Arabic, specifically focusing on the Egyptian Dialect; their NER system identified person and location name entities by applying an Inside I, Outside O and Beginning B (IOB) tagging scheme. The Egyptian dialect dataset was chosen from a set of web blogs containing nearly 40k tokens, including 285 persons and 153 location named entities. The experiment's results show that their system achieved 91% F-score for location NER when applying a Levenshtein match approach to compare the similarities between the input and a gazetteer entry. In (Zirikly and Diab, 2015) the researchers used two datasets to develop an Arabic entity system. The first dataset contains Arabic tweets, and the second one is an Egyptian dialectal dataset proposed by (Zirikly and Diab, 2014). The results show that the performance of the proposed Arabic NER system underperformed when tested on the Twitter corpus. The best F-score result achieved was 61% when applying gazetteers, Brown Clustering (Brown et al., 1992) and Word2vec Cluster (Mikolov et al., 2013). On the other hand, their system achieved a 91% F-score for location NER when tested on a formal Egyptian dialectal dataset.

In this research, in addition to considering the location NER task, it focused on inferring event location, that is, some tweets that mention flood events contain more than one location NE, and the aim is to infer just one location which refers to the flood event location.

## 3.5. Chapter summary

Overall, the review above explain that while several approaches are available for carrying out event detection using social media, these approaches are usually used only on a large scale, for example for large-scale event detection systems, which means they are not suitable for use on the small-scale. Meanwhile, small-scale event detection systems are also available, but these are highly specific and often limited to detecting one off events while overlooking the circumstances around larger events. Furthermore, the use of social media data to predict disasters has led to researchers proposing a number of event detection systems for emergency responses, which attempt to filter, search and analyse tweets throughout disaster events in real time. However, the majority of these approaches do not effectively highlight the event location NE or distinguish it from other location NEs mentioned in the text, such as a tweet, or use a specific language- most often the English language.

Event detection approaches often suffer from limitations with regard to the real-time aspects of the approaches used for identifying events on social media. Therefore, these limitations, along with the limitations presented by traditional approaches to detecting events, provide the motivation and rationale for this research. It is essential to develop a framework that incorporates multiple languages, and that is suitable for both large-scale and small-scale social media content in real-time. It is also necessary to introduce emerging techniques for NER that can find an event's location based on social media texts with a high level of accuracy, which is relevant and provides information on an event.

Therefore, an original approach to event detection is proposed, which should overcome most of the limitations and challenges mentioned above, and which should provide an effective system for detecting large-scale events along with related small-scale events. There are five components to the integrated event detection framework that is proposed, which are: data collection; pre-processing; classification (supervised machine learning algorithms); named entity recognition (natural language processing); named entity linking (knowledge base), and event visualiser (representation). This system would automatically identify the location of floods events in real world time. It would use a classification algorithm that includes a sliding window timeframe that can detect and distinguish real time flood events from social media streams or from previous events described in a tweet.

Therefore, the deployment of a supervised classification model will be used for the classification of each tweet based on an event class or a non-event class before detecting the location, as form of NER. In addition, model's performance will be evaluated, in particular the integrated framework, based on the results from Twitter datasets for several events to assess how effective the framework is. A case study approach will also be used to compare it to other approaches that use Twitter posts, along with rainfall data for the same period of time collected from the National Oceanic and Atmospheric Administration.

This chapter has presented various studies on event detection contained in the literature, over a variety of domains, and has connected these studies to this work on flood detection in real-time. Furthermore, the first Systematic Literature Review (SLR) has utilised a search protocol to strictly summarise the key characteristics of the different TC techniques and methods used to classify Arabic text. The SLR results have revealed that little work has been done on colloquial Arabic text classification, and none of these works have discussed

natural event detection using classification techniques. The next chapter of this thesis will address filling this gap in the literature by producing a comparative study using a real word flood data collected from Twitter. A literature review on the event detection techniques in social media has placed particular focus on new event detection, as well as techniques that have been used in inferring the location NE from social media networks. In addition, the NER approaches that have been applied to Arabic tweets have been discussed, and their limitations and gaps explained.

# Chapter 4: Research Methodology

## 4.1 Research Method

This research explores the reliability of using Twitter to identify flooding events in real time, and it therefore required an agile model of development, along with experimental research and a building research methodology. An experimental research methodology is split into two phases, which are an exploratory phase and an evaluation phase. A build research methodology involves constructing a physical artefact, or software system, in order to check whether it is feasible (Amaral, 2011). It is essential to conduct an exploratory phase in order to discover the hidden parameters of the design being proposed, as these may not be explicit in the general research questions at the start. Furthermore, the evaluation phase can then be used to analyse the impact from varying the parameters.

## 4.2 Proposed approach

Figure 4-1 illustrates a flow diagram that shows the methodology used in the current research. The following subsequent sections provide further detail on what is involved at each stage shown in the flow diagram.

### 4.2.1 Objectives

Firstly, the objectives that need to be met in order to decide whether the project is successful or not need to be presented, as this facilitates a list of research questions being produced to guide the research.

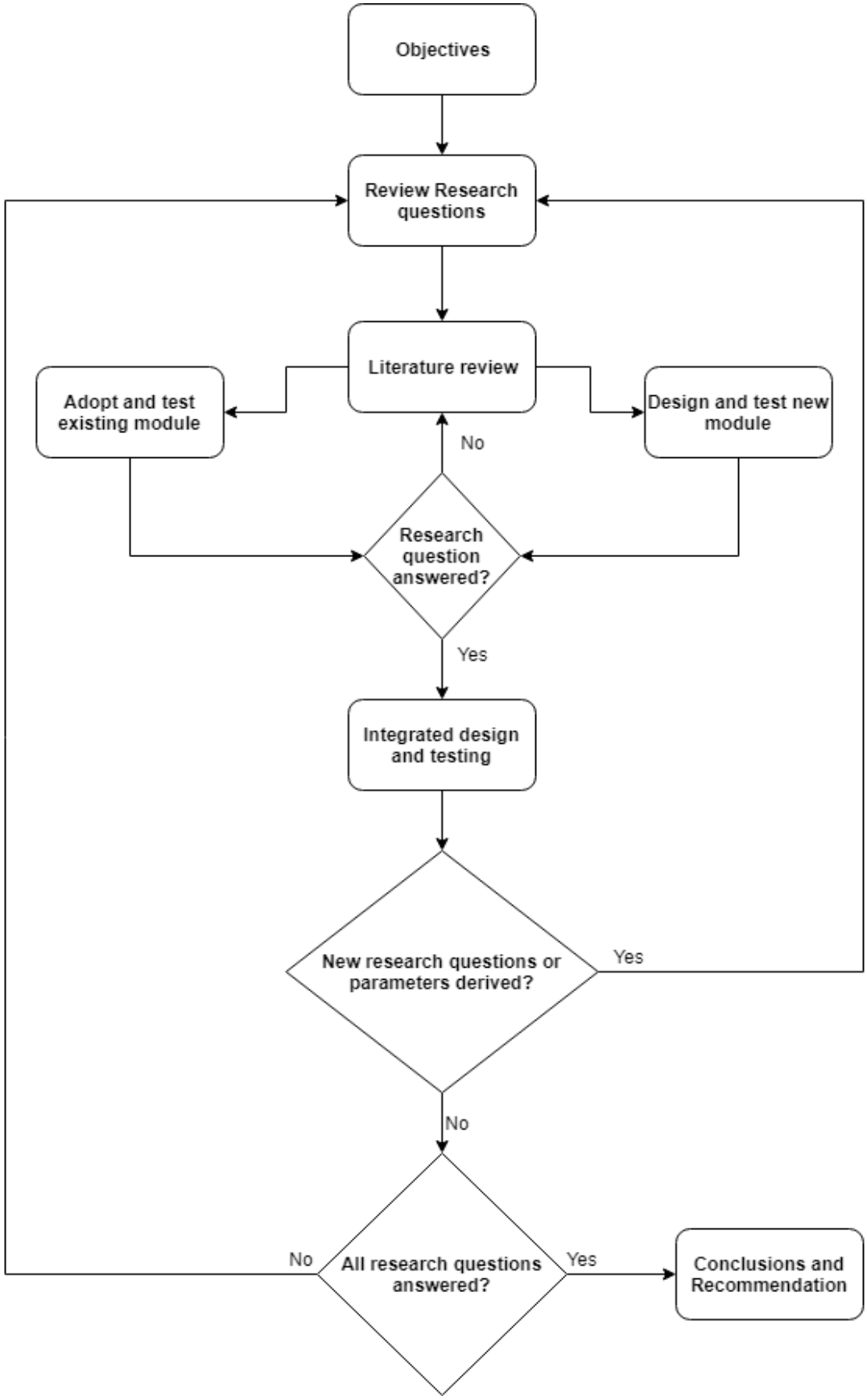


*Figure 4-1 flow diagram of the methodology used in this research*

### 4.2.2 List Research questions

Chapter One contained a broad set of research questions, each of which could possibly have its own additional questions. This initial stage of the research set out in Chapter One allows a broad list of research questions to be devised ready to pass them on to the next phase.

### 4.2.3 Literature review

A literature review has been carried out in order to explore and understand each of the research question, starting by acknowledging what previous researchers have done; the problems that they faced, and what has been explored or neglected in related areas of study. In this way, it should be possible to determine whether any modules exists that can be adopted to solve the current research problem, or if a new module should be designed to address the problem.

### 4.2.4 Adopt & test existing modules

During this stage, the existing modules found in the literature are adopted to see whether they are suitable for the purposes of the current project. Adopting a module may be done by using hardware, software, or both, for example adopting part of an existing algorithm and modifying it for Arabic text if it does not already do so. After modifying the module, it must be tested and documented before progressing to the next step in the methodology.

### 4.2.5 Design and test new module

The designing and testing stage runs parallel to the previous phase, and if the literature reveals that there are not readily available solution to the research question exists, a new solution will be proposed and developed. This new solution must then be tested and documented before moving on to the next phase.

### 4.2.6 Research question answered?

Testing individual modules supports discovering whether the corresponding research question has been answered, and if so, the flow can move on to the next stage; if not, the flow loops back to analyzing the literature in an attempt to find an alternative solution to the previously mentioned branches, and this may involve adopting existing modules or designing new ones.

### 4.2.7 Integrated design and testing

The agile approach used in this research means that integrated design and testing are performed during every iteration where separate modules are integrated, or new functionality is being added. In this way, any additional functionality at later iterations can be added if a correctly functioning framework is achieved.

### 4.2.8 New sub questions or parameters derived?

Individual modules can lead to new sub questions or parameters being necessary, perhaps as a result of integration, or during the design process, but if there are no new questions or parameters at this stage, the flow can continue on down to the next phase. However, if new sub questions or parameters do arise, the flow loops back to the research questions in order to integrate these new questions and parameters.

#### 4.2.9 All research questions answered?

The flow may be directed back to the first stage if only a subset of the questions has been answered so far. Provided that all of the questions have been answered, the flow moves on to the final stage in the methodology.

#### 4.2.10 Conclusions & recommendations

The conclusions must be based on the results obtained during the testing stages, and following the analysis of the overall system and an assessment of whether it has met the research objectives by answering the research questions.

Finally, recommendations for future work can be made, along with suggestions for overcoming any limitations faced while designing the system. The recommendations also state how the project can be extended to involve other work.

### 4.3 System Design

Figure 4-2 below presents a conceptual diagram showing the proposed flood detection system. The system's architecture contains five main components, which are: Micro-blog Loader, Pre-processor, Classifier, Location Detector, and finally Event Visualiser.

#### 4.3.1 Micro-blog Loader: Tweets Loader

Public tweets are obtained by the Micro-blog Loader which has been designed to access Twitter's Application Programming Interface (API) service. An initial query using a set of keywords is used in an attempt to match up with any tweets that may refer to flooding events. In addition, the Micro-blog Loader collects related metadata provided by the API, for

example the time the tweets were published, the language of tweets, geolocation information, and the source and type of tweet.

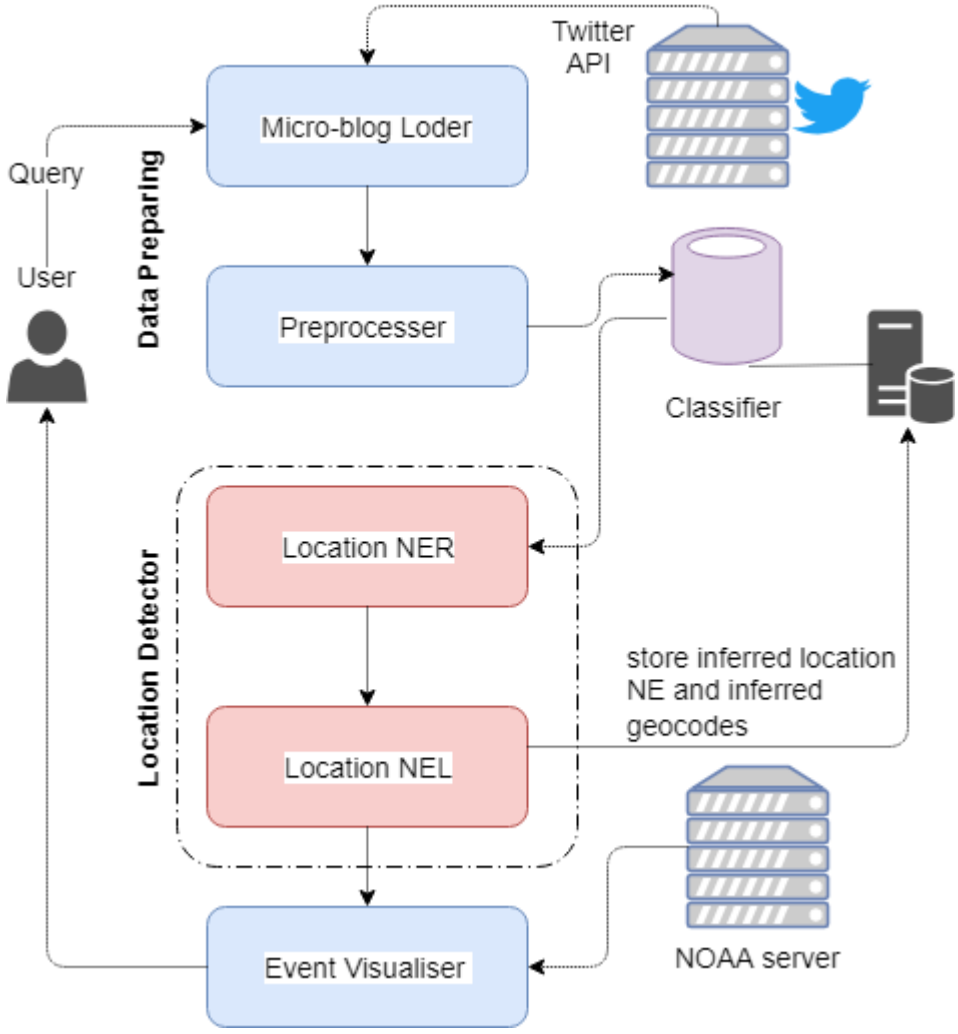


*Figure 4-2 The architecture of the proposed system*

### 4.3.2 Pre-processing

The tweets are filtered by the pre-processor according to specific inclusion and exclusion criteria. The pre-processing of unstructured text, including as tweets, makes text classification easier as it removes any worthless data. The pre-processing stage requires the following steps to be followed:

- Step 1: Remove new line characters, and all punctuation marks or numbers

- Step 2: Remove non-Arabic text characters
- Step 3: Remove diacritics
- Step 4: Normalise letters by, for example, standardising certain letters that are used interchangeably in colloquial Arabic, for example converting (أ،آ) to ( ا).
- Step 5: Exclude words that add no value to the text classification, such as stop-words and prepositions, which are frequently used in tweets.
- Step 6: Calculate the Term Frequency (TF) and Document Frequency (DF) in order to generate the training and testing matrix (TF-IDF matrix).

### 4.3.3 Classifier

The predict function will be used by the classifier to predict a class for the tweets in the testing set. In the current research, Support Vector Machine (SVM) will be used due to experimental results showing that SVM performs better than other classifiers with colloquial Arabic text during pre-processing and under various stemming conditions.

### 4.3.4 Location Detector

To discover the location, the system will examine the metadata associated with all tweets, and it will incorporate Named Entity Recognition (NER) and Named Entity Linking (NEL) to highlight the location of events on an interactive map provided by the Event Visualiser, as shown in Figure 4-2. Once collected, the tweets are stored on a database to facilitate an improved prediction of future flood times and locations. Therefore, data that is saved will be limited to tweets that contain geocoding information and metadata.

### 4.3.5 Event Visualiser

A user friendly Human Computer Interaction (HCI) method is provided by the Event Visualiser, which assists users in locating floods and making decisions about incident responses. The Event Visualiser uses an interactive map that utilises automatic reactive binding between inputs and outputs, which forms the core component of the system. There are also other components such as a form for gathering user input data, including a time-scale for the query (e.g. the last hour, or the time span between two dates or times), as well as a metadata table for illustrating the details of the tweets that are being used. As well as metadata, multimedia attached to the tweets will also be visualized, as multimedia such as images or videos of an event can help in quantifying the risk.

## 4.4 Specification

An overview of the necessary requirements for the systems that will be used to for the implementation, along with the requirements of the implementation itself, will be set out in this section. It will set out the requirements of the system being developed that will be used by the developers to run system. These requirements will help to develop the final design used to produce the system.

### 4.4.1 Software System Attributes

- **Availability:** In the event of the program crashing, the user must be able to restart immediately having lost only what was typed since the last time the program was saved. Thus, it is the user's responsibility to regularly save the data to reduce the risk from the computer crashing.

- **Portability:**

  1. The system runs on Windows.

  2. The system runs on Macintosh systems.

  3. The system runs on Unix systems.

  4. The code is readable and easily maintainable.

  5. Python and R are used to write the system as these are machine independent programming languages.

  6. The GUI may be accessed using an online internet browser.

- **Validity:** Rainfall data collected from reliable worldwide organisations will be used by the system to obtain credible statistics.

- **Completeness**: Each tweet will be linked to geocode coordinates, with the NA value avoided as far as possible.

- **Privacy**: As the data will be collected from the Twitter API, it is considered open data and will not involve sensitive personal information.
- **Accuracy**: Tweets that mention flooding events' locations will be measured for accuracy, and the performance and results of the suggested location will be compared to the actual location using Google services.
- **Usability**: An interactive visualisation platform will be used to produce the maps to ensure usability and good representative data.

### 4.4.2 Hardware Requirements

The software does not require any specific hardware interfaces, and the hardware device that will be used to develop system model is an ASUS computer that uses Windows 10

### 4.4.3 Software Requirements

The following software applications will be used to develop the system model:

- Python version 2.7: The Python interpreter available from www.python.org will be used for the system and to interpret all Python software. The developer, as well as the user, will use the Python software to write all of the programs. This software interface must be installed on the computer, and the "python <filename>" command must be used to make the program work.
- Version 3.1 of the R language will be used to run the application, as R is a language and environment used to carry out statistical computing and for graphics. It contains a range of graphical and statistical techniques, such as classical statistical tests; linear and nonlinear modelling; time-series analysis; classification, and clustering. R is highly extensible and is available as free software according to the terms of the Free Software Foundation's GNU General Public License in source code form. It is made up of and runs on a wide variety of UNIX platforms; FreeBSD; Linux; Windows, and MacOS.

### 4.4.4 System Analysis and Design Tools

The tools and techniques that will be used for the system design and analysis are presented in Table 4-1 below:

*Table 4-1 System analysis and design tools.*

| Tool | Description |
|---|---|
| **Draw.io** | A free online diagram software for making flowcharts, process diagrams, org charts, UML, ER and network diagrams. Draw.io was used for designing system phases modules and designing visualization prototypes. (https://www.draw.io/ ) |
| **dataturks** | A free online data annotations tools. (https://dataturks.com/ ) |

### 4.4.5 System Implementation Tools

The tools and techniques that were used for the system's implementation are shown in Table 4-2:

*Table 4-2 System implementation tools.*

| Tool | Description |
|---|---|
| **Twitter API** | The Twitter's standard search API (search/tweets) allows simple queries against the indices of recent or popular Tweets and behaves similarly to, but not exactly like the Search UI feature available in Twitter mobile or web clients. The Twitter Search API searches against a sampling of recent Tweets published in the past 7 days. ( Twitter API, 2018) |
| **Google Geocoding API** | The Geocoding API is a service that provides geocoding and reverse geocoding of addresses.<br>**Geocoding** is the process of converting addresses (like a street address) into geographic coordinates (like latitude and longitude), which you can use to place markers on a map, or position the map.<br>**Reverse geocoding** is the process of converting geographic coordinates into a human-readable address.<br>You can also use the Geocoding API to find the address for a given place ID. (Google, 2018a) |
| **arabicStemR** | It is a R package for stemming Arabic for text analysis.<br>(https://cran.r-project.org/package=arabicStemR ) |
| **R Markdown** | Convert R documents into a variety of formats including HTML, MS Word, PDF, and Beamer. We integrate R and Python using R Markdown to have the flexibility of using and transferring data between both environments.<br>(https://cran.r-project.org/package=rmarkdown ). |
| **R Shiny** | Shiny is an R package that makes it easy to build interactive web apps straight from R. You can host standalone apps on a webpage or embed them in R Markdown documents or build dashboards. You can also extend your Shiny apps with CSS themes, html widgets, and JavaScript actions.<br>(https://shiny.rstudio.com) |

| | |
|---|---|
| **The rnoaa package** | Rnoaa is an R package, many functions in this package interact with the National Climatic Data Center application programming interface (API) at https://www.ncdc.noaa.gov/cdo-web/webservices/v2 . An access token, or API key, is required to use all the ncdc functions. (https://cran.r-project.org/web/packages/rnoaa/rnoaa.pdf ) |
| **The tm package** | tm (shorthand for *T*ext *M*ining Infrastructure in R) provides a framework for text mining applications within R. The tm package offers functionality for managing text documents, abstracts the process of document manipulation and eases the usage of heterogeneous text formats in R. The package has integrated database back-end support to minimize memory demands. An advanced meta data management is implemented for collections of text documents to alleviate the usage of large and with meta data enriched document sets. (http://tm.r-forge.r-project.org/ ) |
| **Leaflet for R** | Leaflet is one of the most popular open-source JavaScript libraries for interactive maps. It's used by websites ranging from The New York Times and The Washington Post to GitHub and Flickr, as well as GIS specialists like OpenStreetMap, Mapbox, and CartoDB. This R package makes it easy to integrate and control Leaflet maps in R. (https://rstudio.github.io/leaflet/ ) |

## 4.5 Data Sets

The following subsections describes the various datasets that will used in the research.

### 4.5.1 Social Networks APIs

In order to build different applications and software, it is useful to use an application programming interface (API), as this contains the procedures, protocols, and tools necessary

to do so. APIs are offered to users by social network platforms so that they can develop web applications, and its programming structure is highly useful for creating new features and adding them to websites (Bowden, 2014). The key components of an API are an operating system; a web-based system, and sometimes a database tool; in addition, APIs use a particular programming language. Therefore, APIs are useful for developing applications that are suitable for various systems. APIs can be used for the GUI components, as well as to gain access to computer hardware such as the hard disk driver. Researchers can gain access to instant data, user activities, and popular topics by using APIs. The next section will provide some background information on the Twitter API, and the datasets that will be collected from Twitter throughout the research.

#### *4.5.1.1. Twitter API*

The Twitter API enables certificated users to search for information in different ways (API Twitter, 2018), and there are four main "objects" on the API that can be used, which are *Tweets*, *Users*, *Entities* (or even *Entities in Objects*), and *Places*. In addition, the API usually includes *Twitter IDs* or *Place Attributes,* and the majority of the people use *oAuth* to gain access to the Twitter API. Requesting a user's signature is decided according to the identity of their application, along with the user's access to the identity granted by the end user, with the user's access token represented by the interface. A *keyword* allows the API to obtain information related to this keyword from tweets from all over the world, and by using *locations,* users can search for other users and their posts and information, including being able to focus on a specific city or place. In addition, *following* another user allows all of their tweets, retweets, and replies to be searched.

However, users cannot do whatever they want with the Twitter API, and there is a limit set for its API in order to prevent possible damage to the bandwidth from hackers or spammers. Thus, users are allowed 180 requests maximum over a 15 minutes period, and this restriction is for obtaining requests rather than posting. If this limit is exceeded, the document produced by a REST call will inform users and the response will be to whitelist them.

In addition, the Twitter API only returns a maximum of 3200 states, no matter how many page counts or parameters there are. Twitter also recommends certain additional restrictions, such as using page attributes, and reducing the count properties. Furthermore, it recommends for results to be saved to a local cache, rather than repeating the request for the same state. Generally, two different forms of HTTP request, POST, and GET exist, and these two forms can invoke the Twitter API. That is, it forwards the POST and GET requests sent by clients to the original API address, and returns the HTTP header and the content back to the client, which covers all of the features of the original Twitter API. While the client may provide an alternatively configured API address, they are not required to change any of the code. The following scenarios show the method most often used by researchers to obtain information by accessing twitter.com directly to view a Friends list (a GET request).

### 4.5.2 The National Climatic Data Center API

NOAA's National Centers for Environmental Information (NCEI) facilitates accessing one of the most important archives in the world, and Climate Data Online (CDO) allows free access to NCDC's archives on global weather and climate data, as well as providing station history information. This historical data includes daily, monthly, seasonal, and yearly measurement data on temperature, precipitation and wind, in addition to radar data and

thirty years old climate averages. All of this data is quality controlled, and it is possible to order most of it in the form of certified hard copies for legal use.

NCDC's Climate Data Online (CDO) has web services available that also give access to current data. Developers can use the CDO API to make scripts or programs which use the CDO weather database and climate data. To use the API, an access token is needed, with every token limited to five requests per second and/or 10,000 requests a day.

The current study will use precipitation data archived at NOAA's National Climatic Data Center, from the time series of daily figures from NOAA cooperative reporting stations located around the Arabian Peninsula. The extent of the data recorded differs from one station to another, and some records only go back as far as 2017.

### 4.5.3 Data Collection

The first dataset is made up of approximately 230975 unique Arabic tweets from the period the 5th of May 2017 to the 1st of June 2017 using Twitter's Streaming API, as this facilitates subscribing to a continuous live stream of new data. The initial aim is to monitor and analyse floods events that occur in a specific region. The floods events can be defined through specific keyword queries, which in Arabic are: "سيول", "فيضانات", "سيل" and "أمطار قوية". This period of time should be useful as heavy rainfall events occurred in Saudi Arabia, and a lot of people were reporting and discussing flooding events. In addition, specific keywords that were often used by Twitter users to report flooding or a heavy rainfall event were chosen. The second dataset represent the rainfall data for Makah region (geocode coordinates located within rectangle bounded by (22.00,39.60) from the northwest and (21.04,41.59)

from the southeast) for the same period of time, this rainfall data were collected from NOAA's National Climatic Data Center.

## 4.6 Experiments & Analysis

Unit tests are performed at preliminary iterations of the agile process to ensure that fundamental modules work properly prior to addition of more functionality. Subsequent iterations with added functionality lead to integration testing.

### 4.6.1 Experiment 1: Systematic Literature Review

The systematic literature review will highlight the performance of various classifiers, as well as the best text pre-processing and Dimensionality Reduction Techniques for Arabic text classification tasks. The key phases used in the systematic literature review are: planning, search strategy, data extraction, and data synthesis.

### 4.6.2 Experiment 2: Classification of Colloquial Arabic Tweets in real-time to detect high-risk floods

The detection of 'floods' as the specific type of high-risk natural disaster explored, will be carried out by classifying informal, that is, colloquial, Arabic text. Although this work is aimed at event detection, it will focus on remote regions where people use colloquial Arabic to tweet.

### 4.6.3 Experiment 3: Location Inference from Twitter

Location inference from Twitter will be used to explore the best way of inferring an event location for a specific type of high-risk natural disaster- flooding- through the application of an Arabic Named Entity Recognition (NER) technique involving colloquial Arabic

text, as although colloquial Arabic is often used on social media, it has not received much attention.

### 4.6.4 Experiment 4: Locations Named Entity Linking

Locations named entity linking is the method proposed for linking and mapping location NEs extracted from tweets to maps by using a free geocoding service from Google. As the aim is for the system to determine and link the location stated in colloquial Arabic tweets, a knowledge base for location NE is also needed, and by applying an entity link, it should be possible to identify the location associated with the location NE mentioned in the in tweets accurately.

# Chapter 5: Classification of Colloquial Arabic Tweets in real-time to detect high-risk floods

## 5.1 Introduction

Twitter remains one of the most popular social media platforms to date. This is evidenced by the huge and increasing real-time texts posted by users on a day-to-day basis. Moreover, the correlation of this data may translate into a wealth of information (Medvet and Bartoli, 2012). Twitter's popularity and features (e.g. available APIs and metadata) have changed traditional offline knowledge discovery methods in favour of data-driven science, focusing on online freely-available unstructured data. Twitter revolves around the concept of microblogging, allowing users to post short texts, with the option to include a link to a website, photos or videos.

This chapter investigates the detection of a specific type of high-risk natural disaster, namely 'floods', by classifying informal (colloquial) Arabic text, which has received very limited attention to date, as shown in section 3.3.2. However, the purpose of this work is event detection, with a focus on remote regions where people tweet in colloquial Arabic. Those who are living in remote rural areas are usually very close to event as they happen, which means they would be able to sense and report events first and in real-time. Therefore, early critical evidence of a specific natural disaster, such as flooding, could come from updates and tweets in colloquial Arabic. We argue in favour of further contribution to this research area, and consider a specific case study to investigate different aspects of event detection based on a corpus consisting of informal Arabic tweets.

## 5.2 Problems and Challenges

Emergencies related to high-risk events require exceptionally fast incident response and decision-making based on first-hand information, and Twitter has eased information flow for decision makers and enabled Open-source Intelligence (OSINT). Short messages posted on social media (tweets) can typically reflect these events as they happen. For example, (Kryvasheyeu et al., 2016) discuss how tweets' analysis outperformed traditional methods used by formal agencies, such as the US Federal Emergency Management Agency (FEMA), in estimating the location and severity of damage by Hurricane Sandy. People are using Twitter to report real-life events, which gives researchers the opportunity to design and develop quality event detection systems. For instance, (Sakaki et al., 2010) have developed an earthquake reporting system in Japan by monitoring and filtering English and Japanese tweets; their system detected earthquakes much faster than the Japan Meteorological Agency (JMA).

While those approaches towards event detection may be customised to various case studies, this work focuses on flood detection in real time. Recently, floods have been the most frequent natural disaster in the Arabian Peninsula; they occur several times a year and cause severe damage to both life and property. In one of the worse cases in 2009, over 120 civilians were reported to have been killed (Al-Saggaf, 2012). Part of the reason for such severe damage is due to the lack of real-time information on events, and inefficient Decision Support Systems (DSS) to utilise the available resources more effectively (Al-Saggaf, 2012); (Al-Saud, 2010).

Many researchers have used machine learning and data mining techniques and models, but not limited to supervised learning, unsupervised learning, NER and NLP for the purpose of identifying specific events occurring on social media, mostly based on the English language;

however, to the best of our knowledge, and further to the very limited number of research on colloquial Arabic in general, no research has been found that addresses flood detection based on Arabic tweets. Therefore, the performance of machine learning methods and techniques has been evaluated for colloquial Arabic text collected and classified directly from Twitter, with the intention of answering the following questions: (i) which of the supervised learning classification algorithms can more accurately outperform others while detecting high-risk floods from colloquial Arabic tweets? And (ii) to what extent could f techniques be used to enhance classification?

Accuracy, recall, F1-measure, and precision have been reported to measure the quality of the classifiers tested. In addition to the above measures, McNemar's test will be used to calculate statistical significance. This has been used in various studies in the area of machine learning, and has proved to be reliable for determining whether the best classifier is significantly better than others (Dietterich, 1998); (Bostanci and Bostanci, 2013).

The remainder of this chapter covers the experiment's methodology, in section 5.3, which also discusses the data collection and extraction from Twitter, data labelling, text cleaning, building a Term Frequency–Inverse Document Frequency (TF-IDF) matrix, training models, and finally, classification of documents. Section 5.4 provides a discussion and presents a visualisation of the classifiers' performance.

## 5.3 Methodology

This section presents the methodology used to classify Arabic tweets to determine events related to high-risk floods. Although there are several data mining tools available such

as Weka (Eibe Frank, 2016), RapidMiner (rapidminer, 2018), and Orange (orange, 2017), R tools were selected for the data analysis since all of the required classification algorithms are integrated into R. R provides a free language and environment for statistical computing and graphics to expedite the text classification process: everything from tweet extraction to training and classifying has been supported to make machine learning with textual data more accessible (Team, 2013). R has the ability to extract and manipulate tweets from Twitter automatically using the TwitterR package (Gentry et al., 2016). The core functions of the program will be discussed and the use of R packages with the relevant coding examples will be demonstrated.

The key phases in the methodology are illustrated in Figure 5-1. As an overview, it has identified how to connect and access Twitter Streaming API, and how to search for queries and how to remove retweets, which formed part of the data collection stage. In the second phase, data labelling was performed to distinguish between event related and non-event tweets. To mitigate bias, the text pre-processing stage has been applied to remove noise, and this has helped to clean the dataset and reduce the size.

In the subsequent phase of this experiment, the dataset was divided into train classifiers and build the Term Frequency–Inverse Document Frequency (TF-IDF) matrix. After the classifiers were dealt with using the training dataset, four experiments were designed to test the performance of each classifier. Further discussion and technical reflection on each of the phases will be provided in the subsequent sections.

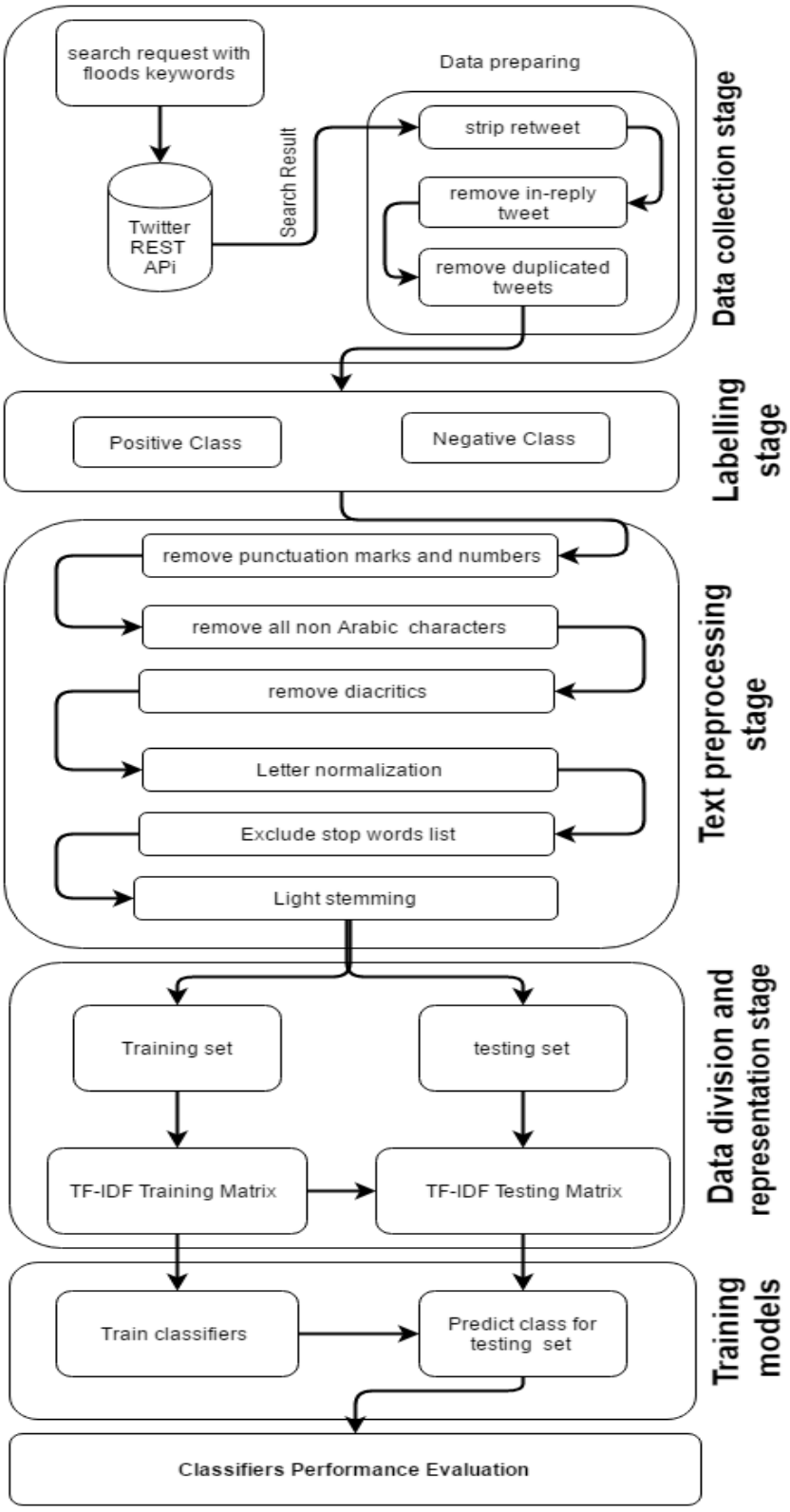


*Figure 5-1 MAIN STEPS IN TWEETS CLASSIFICATION*

### 5.3.1 Data Collection

Data collection has been conducted with the help of Twitter REST APIs (Twitter, 2018c), which can be accessed using Twitter user credentials via Open Authentication (OAuth). The data returned from the APIs includes internal data that refers to unstructured data in the tweet content, that is, the text of the tweet itself, and external data that refers to the structured data behind tweets such as tweet ID, retweets, in replay to user, tweet language and tweet location. The content of the internal data was used to learn about and test the classifiers, while the external data was used to clean and prepare the corpus. Only tweets directly mentioning 'floods' or 'torrents' ('سيول OR فيضانات') in Arabic were collected, and only original tweets have been included. Retweets were removed, as the focus of this work is on real-time content published for the first time. Duplicates and in-reply tweets were also removed from the data. Algorithm 1 demonstrates the implementation in more detail:

**Algorithm 1**

***Input***:

$A$ = {$term_1$, $term_2$ ,…, $term_i$,..., $term_n$} ; each term represent the key words related to the floods in Arabic.

***Output***:

Data frame contains returned tweets and its meta data.

***Data collection steps:***

**Step 1 :** Connect with Twitter API using the *setup_twitter_oauth* function.

**Step 2 :** Issue a search request using the *searchTwitter* function to collect the tweets based on the key words in $A$.

**Step 3 :** Remove retweeted tweets from the Twitter API respond result using the strip_retweets function to provide a pure set of tweets.

**Step 4 :** The outcome from searchTwitter function is a list; converts the list into a data frame named DF by using *twListToDF* function.

**Step 5 :** Remove in-reply tweets from DF.

**Step 6 :** Remove duplicated tweets from DF.

***return*** DF data frame

### 5.3.2 Data Labelling

The supervised learning of classification algorithms requires manual data labelling. To facilitate this, a new column ***Event_Class*** was inserted into the data collected to indicate the classification of tweets as positive or negative. A positive class includes tweets describing high-risk floods. These would typically cause damage to infrastructure (railways, villages, towns, industrial plants, etc.), occur in areas where causalities have been reported in earlier years, or described as damaging in the tweet itself. In addition, a negative class would include any tweet mentioning floods, but that could not be labelled or referred to as a high-risk event for any of the following reasons:

- The tweet is discussing a non-real time event (floods or torrents that occurred in the past).
- The tweet could be discussing the occurrence of a weak and therefore low-risk flood. Therefore, no damage or service disruptions (e.g. transport disruption) would be expected.
- A tweet including relevant keywords (flood/ torrent) but in a different context (e.g. poems, jokes, etc.).

Three different annotators were asked to manually label 1500 randomly selected tweets, and divide them into two classes, in order to test the classifier. These classes are 'Event' and 'Non-Event.' This resulted in 1434 tweets that were suitably for classified, as all three annotators placed them in the same class. Prior to the classification, the annotators were given a set of instructions and some examples to make the annotation task clear; these instructions are shown in Table 5-1 below:

*Table 5-1 Instructions for annotators prior to the annotation task (classification)*

| |
|---|
| Instructions: Take a Twitter message, identify whether the message is an:<br>(A) Event or (B) Non-Event<br>Please read the examples and the invalid responses before starting if this is your first time working on this annotation task. |
| Tweet: ("الأرصاد": أمطار على نجران وما جاورها حتى التاسعة مساء https://t.co/twQbs3Eu36 sabqorg #سبق)<br><br>Overall the tweet is: ☐ Event ☐ Non-Event |

Some of the example tweets and annotations given to the annotators are shown in Table 5-2 below:

*Table 5-2 Example tweets and annotations to the annotators before the classification task (Classes are: Event or Non-Event).*

| Example tweets | Event or Non-Event |
|---|---|
| "الأرصاد": أمطار على نجران وما جاورها حتى التاسعة مساء # سبقhttps://t.co/twQbs3Eu36 sabqorg | Non-Event |
| 29أسرة تضررت من سيول جازان.. والمدنيّ تحصر المنازل #واس #خبر #السعودية #اخبارhttps://t.co/Lm779daeSK | Non-Event |
| الحمدلله لا فيضانات ولا سيول ولا زلازل وبراكين نعمة تستحق الشكر والثناء وعوضنا بالصبر وبالجنة ونعيمها #الجو_يقول_لنا | Non-Event |
| سيول جارفة بمناطق عدة بمديرية نصاب في #شبوة | Event |
| سيول جارفة داهمت شارع شهار في الطائف وجرفت عدد من المركبات عصر اليوم https://t.co/nXZTKZ3Lbz | Event |
| شاهد سيول الطائف: غرق شوارع وانجراف سيارات وجدران # شيء_يهزكhttps://t.co/fQffrONcOd | Event |

### 5.3.3 Text pre-processing

Unstructured text such as tweets requires pre-processing before it can be analysed. Pre-processing is actually a trial to improve text classification by removing worthless data. Due to this, a text mining (tm) package (Feinerer et al., 2015) and Arabic stemmer arabicStemR package (Nielsen, 2017) were installed. The arabicStemR package contains functions to allow the stemming and cleaning of Arabic texts. Algorithm 2 below were implemented as part of the pre-processing stage.

**Algorithm 2**

***Input***:

DF data frame from Algorithm 1

***Output***:

Data frame contains tweets and its meta data after pre-processing sptes.

***Pre-processing steps:***

***for*** ∀ $tweet_i \in DF$ ***do:***

**Step 1 :** Remove new line characters.

**Step 2 :** Remove punctuation marks.

**Step 3 :** Remove numbers.

**Step 4 :** Clean all characters that are not Latin or Arabic.

**Step 5 :** Clean Latin characters.

**Step 6 :** Remove diacritics from Arabic unicode text.

**Step 7 :** Reduction of the number of words through standardising different Hamza on Alif seats (أ,إ and آ converted to ا).

**Step 8 :** Exclusion of words adding no value to the Text Classification scheme (stop-words) such pronouns and prepositions, which frequently occur in all tweets.

***return*** DF data frame

In this study, four experiments have been implemented with different criteria to evaluate the impact of the stemming process on colloquial Arabic text, as shown below:

***Experiment A***: Colloquial Arabic text without stemming.

***Experiment B***: Colloquial Arabic text with Light 10 stemmer.

```
tweets.df2$text = sapply(tweets.df$text,
                         function (x)  doStemming(x)$text)
```

***Experiment C***: Colloquial Arabic text with the removal of common prefixes.

```
tweets.df3$text = sapply(tweets.df$text,
                         function (x)  removePrefixes(x))
```

***Experiment D***: Colloquial Arabic text with the removal of common suffixes.

```
tweets.df3$text = sapply(tweets.df$text,
                         function (x)  removeSuffixes(x))
```

In linguistic morphology, stemming is typically performed on datasets to reduce inflected words to their word stem or base. In this experiment, this step has not been used- not to produce the linguistic root of a given Arabic surface form, but to remove the most frequent suffixes and prefixes. Common prefix or suffix strings such as ال، بال، ان، ات، ة have been removed. The latest Light 10 Stemmer was used because it outperforms other stemmer techniques, especially with colloquial Arabic text (Al-Badarneh et al., 2016) (Al-Wehaibi and Khan, 2015).

### 5.3.4 Data Division

The number of tweets after the pre-processing stage was 1434 tweets. The corpus was divided into two sets- one for training (70% of the data) and the rest for testing purposes (30% of the data) using the createDataPartition function (Kuhn, 2008), which can be used to preserve the distribution of the classes in the training and test sets, as shown in Table 5-3.

```
selected <- c("EventClass","text")
tweets.df <- tweets.df[selected]
indexes <- createDataPartition(tweets.df[,1] , p = 0.7, list = FALSE)
train.data <- tweets.df [indexes,]
test.data <- tweets.df [-indexes,]
```

Outlining training data is inevitable to build the classifiers and fit the parameters. On the other hand, test data were used to compare the performance of the classifiers that were created based on the training set. By the end of this process, the words were ready for indexing and for calculating the TF-IDF weight.

*Table 5-3 Data division*

| *Set* | *Positive class* | *Negative class* | *Total* |
|---|---|---|---|
| *Training set* | 315 | 688 | 1003 |
| *Testing set* | 136 | 295 | 431 |
| *Entire set* | 451 | 983 | 1434 |

### 5.3.5 Data Representation

The classifiers cannot directly use text. Instead, it is a requirement to first extract and generate the frequency list of the dataset features (single words, light stemming, words without prefixes and words without suffixes). After that, TF (Term Frequency) and DF (Document Frequency) are calculated to generate the training and testing matrix cells using the TF-IDF weighting method. TF-IDF assigns higher weights to distinguish the terms in a document. In other words, the more a term occurs in a document, the more it is representative of the content of the document. Moreover, the more the documents contain the term, the less informative it becomes (Paralic and Bednar, 2003). Figure 5-2 shows 5 examples of tweets and show the changes at pre-processing, stemming, data representation (TF-IDF) stages.


| | احد | الاستخلاص | الباحه | البهيته | الخميس | الزملا | السيارات | السيل | العنامبه | الكبير | المنستير | المياه | اليوم | بلاد | بهرقله | بيده | تجرف |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| جمال المنستير فيضانات عارمه تجرف السيارات فيديو | 0 | 0 | 0 | 0 | 0 | 0 | 0.33 | 0 | 0 | 0 | 0.33 | 0 | 0 | 0 | 0 | 0 | 0.33 |
| سوسة سيول جارفة المياه تغمر محطة الاستخلاص بهرقلة فيديو | 0 | 0.26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.26 | 0 | 0 | 0.26 | 0 | 0 |
| سيول هادره بلاد طويرق منقول احد الزملا | 0.33 | 0 | 0 | 0 | 0 | 0.33 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.33 | 0 | 0 | 0 |
| جانب سيول طريق السيل الكبير جهه البهيته عصر اليوم الخميس | 0 | 0 | 0 | 0.23 | 0.23 | 0 | 0 | 0.23 | 0 | 0.23 | 0 | 0 | 0.13 | 0 | 0 | 0 | 0 |
| صور سيول الغنامبه بيده شمال الباحة عصر اليوم رصد | 0 | 0 | 0.26 | 0 | 0 | 0 | 0 | 0 | 0.26 | 0 | 0 | 0 | 0.15 | 0 | 0 | 0.26 | 0 |

| | تمر | جارفة | جانب | جمال | جهه | رصد | سوسة | سيول | شمال | صور | طريق | طويرق | عارمة | عصر | فيديو | فيضانات | محطة | منقول | هادره |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| جمال المنستير فيضانات عارمة تجرف السيارات فيديو | 0 | 0 | 0 | 0.33 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.33 | 0 | 0.19 | 0.33 | 0 | 0 | 0 |
| سوسة سيول جارفة المياه تغمر محطة الاستخلاص بهرقلة فيديو | 0.26 | 0.26 | 0 | 0 | 0 | 0 | 0.26 | 0.04 | 0 | 0 | 0 | 0 | 0 | 0 | 0.15 | 0 | 0.26 | 0 | 0 |
| سيول هادره بلاد طويرق منقول احد الزملا | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.05 | 0 | 0 | 0 | 0.33 | 0 | 0 | 0 | 0 | 0 | 0.33 | 0.33 |
| جانب سيول طريق السيل الكبير جهه البهيته عصر اليوم الخميس | 0 | 0 | 0.23 | 0 | 0.23 | 0 | 0 | 0.03 | 0 | 0 | 0.23 | 0 | 0 | 0.13 | 0 | 0 | 0 | 0 | 0 |
| صور سيول الغنامبه بيده شمال الباحة عصر اليوم رصد | 0 | 0 | 0 | 0 | 0 | 0.26 | 0 | 0.04 | 0.26 | 0.26 | 0 | 0 | 0 | 0.15 | 0 | 0 | 0 | 0 | 0 |

*Figure 5-2 examples of tweets and the changes at pre-processing, stemming, data representation (TF-IDF) stages*

### 5.3.6 Training Models

In this step, the training matrix that contains the selected terms and their corresponding TF-IDF weights in each tweet of the training data was used to train the classification algorithms by learning the characteristics of every class from a training set of

tweets. The training process constructs a classification model for testing. In training stage, the most used classification algorithms to classify Arabic text in literature are applied, as mentioned in Table3-7 (section 5-3) the most used classifiers for Arabic text classification are SVM, DT, NNET, NB and k-NN.

1. Support Vector Machine (SVM)

```
svm.classifier <- svm(train.dtMatrix, as.factor(train.data[,1]))
```

2. Decision Tree (j48)

```
J48.classifier   <-    train(train.dtMatrix, as.factor(train.data[,1]),
                                  method = "J48")
```

3. Decision Tree (C5.0)

```
C5.classifier <- C5.0(train.dtMatrix, as.factor(train.data[,1]))
```

4. Neural Networks (NNET)

```
nnet.classifier  <- train(train.dtMatrix, as.factor(train.data[,1])),
                          method = "pcaNNet")
```

5. Naive Bayes (NB)

```
nb.classifier <- naiveBayes(train.dtMatrix,
                                  as.factor(train.data[,1]))
```

6. K-Nearest Neighbor (k-NN)

```
kknn.model <- train(train.dtMatrix,
                          as.factor(train.data[,1]), method = "kknn")
```

After that, the classification model consequently predicts a class for tweets in the testing set using the predict function. The same terms that were extracted from the training data and the same weighing methods were used to test the classification model.

```
predict(classifier_name, test.dtMatrix)
```

## 5.4 Results and discussion

The experiments performed included a total of 431 test tweets to be classified against two categories. Table 5-4 shows the accuracy, precision, recall, and F1-measure results that were obtained when running the selected classification algorithms on the processed corpus using four testing experiments. As shown in Table 5-4, the SVM classifier achieved the highest F1-measure in experiment A, with 93.3%. The second highest F1-measure in experiment A equals 89.2% and it was achieved using the C5.0 classifier, while the performance of the NB classifier scored the lowest F1-measure with 81.8%. A cross-comparison of the data produced in each experiment reveals that experiment A, testing colloquial Arabic without stemming, achieved the highest F1-measure, as shown for the SVM, J48, C5.0 and NB classifiers. However, the removal of the common prefix in Experiment C increased the performance of the classifiers NNET and k-NN at 1.2% and 0.6% respectively. However, it decreased the F1-measure value to 3.2% for SVM.

Recall values address the following question: 'Given high-risk floods event, will the classifier detect it?' On the other hand, Precision Values answer the following question: 'Given a positive prediction from the classifier, how likely is it to be correct?' Table5-4 shows that NNET achieved the highest recall value in experiment C at 94.5%. However, SVM, J48, C5.0 and NB classifiers outperformed NNET in term of precision, which indicates more false positives for the NNET classifier. Hence, the F1-measure was calculated, which is the harmonic mean of Recall and Precision, as demonstrated earlier in Section 2.4.

*Table 5-4 Algorithm accuracy, precision, recall and F-score*

| | Algorithm | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|
| Experiment A | SVM | 0.907 | 0.959 | 0.910 | 0.933 |
| | J48 | 0.837 | 0.945 | 0.833 | 0.885 |
| | C5.0 | 0.846 | 0.946 | 0.844 | 0.892 |
| | NNET | 0.830 | 0.813 | 0.930 | 0.867 |
| | NB | 0.716 | 0.935 | 0.728 | 0.818 |
| | k-NN | 0.781 | 0.745 | 0.920 | 0.823 |
| Experiment B | SVM | 0.860 | 0.955 | 0.857 | 0.903 |
| | J48 | 0.842 | 0.884 | 0.884 | 0.884 |
| | C5.0 | 0.832 | 0.874 | 0.880 | 0.876 |
| | NNET | 0.801 | 0.803 | 0.897 | 0.847 |
| | NB | 0.693 | 0.867 | 0.733 | 0.794 |
| | k-NN | 0.773 | 0.743 | 0.902 | 0.814 |
| Experiment C | SVM | 0.856 | 0.969 | 0.843 | 0.901 |
| | J48 | 0.835 | 0.935 | 0.841 | 0.885 |
| | C5.0 | 0.839 | 0.894 | 0.874 | 0.883 |
| | NNET | 0.842 | 0.816 | 0.945 | 0.875 |
| | NB | 0.714 | 0.911 | 0.734 | 0.812 |
| | k-NN | 0.784 | 0.769 | 0.900 | 0.829 |
| Experiment D | SVM | 0.891 | 0.949 | 0.894 | 0.920 |
| | J48 | 0.839 | 0.918 | 0.857 | 0.886 |
| | C5.0 | 0.846 | 0.932 | 0.856 | 0.892 |
| | NNET | 0.798 | 0.766 | 0.926 | 0.838 |
| | NB | 0.71 | 0.878 | 0.744 | 0.805 |
| | k-NN | 0.763 | 0.739 | 0.897 | 0.810 |

Figure 5-3 illustrates the performance of the classifiers within the conditions of all four experiments; it also shows that light stemming has a negative impact on the accuracy of the classifiers. Light stemming was designed to reduce the index terms by removing a set of prefixes and suffixes, without trying to find roots. Light stemming is mainly dependent on an understanding of Arabic morphology (Paralic and Bednar, 2003); this study covers informal and dialectal Arabic text (non-standard orthography, vocabulary, morphology, and syntax), which could explain the reasons behind the errors produced by the light stemming algorithm.

In section 3.3.2, SVM was reported by several papers to outperform other classifiers with MSA text. Figure 5-3 confirms SVM to be the most accurate method for the classification of colloquial Arabic text as well.

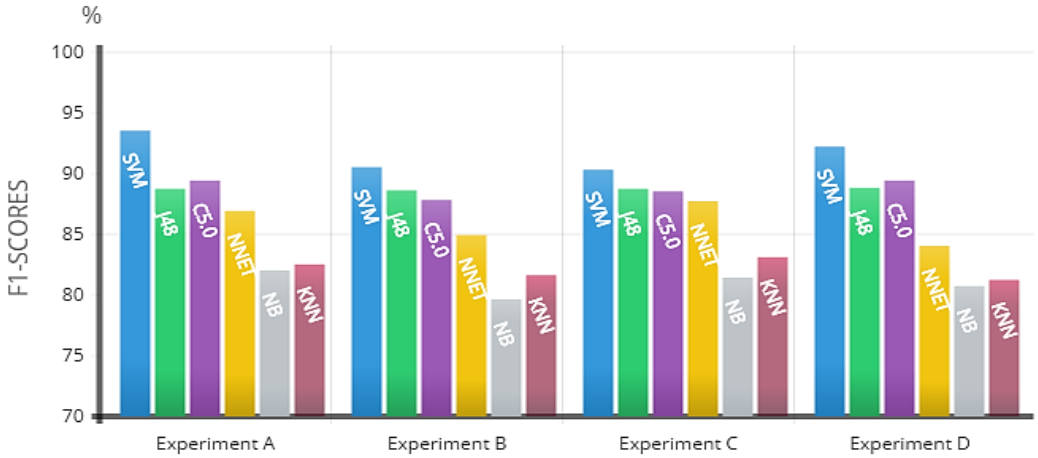


*Figure 5-3 Classifiers F1-Scores*

Furthermore, McNemar's test was applied to the test set of the SVM classifier for experiments A and B to determine whether there is any significant difference between using a light stemmer or work without stemming. Tweets from the test set that have been classified by SVM have also been recorded for both experiments, as shown in Table 5-5.

*Table 5-5 Classification results of classifiers*

| | **Classifier B** | |
|---|---|---|
| **Classifier A** | $n_{00} = 28$ | $n_{01} = 32$ |
| | $n_{10} = 12$ | $n_{11} = 359$ |

Where $n_{00}$ refers to the number of tweets misclassified by both classifiers, $n_{01}$ is the number of tweets misclassified by classifier A but not by classifier B, $n_{10}$ is the number of tweets misclassified by classifier B but not by classifier A, and $n_{11}$ is a number of tweets correctly classified by both classifiers. The null hypothesis ($H_0$) for this test suggests that classifiers A

and B perform similarly, whereas the alternative hypothesis ($H_1$) claims otherwise, suggesting that the classifiers perform differently; a low value of the calculated p-value (< 0.05) could be considered a significant result, rejecting the null hypothesis.

McNemar's test for calculating chi-square using an equation is shown below:

$$x = \frac{(|n_{01} - n_{10}| - 1)^2}{(n_{01} + n_{10})} \quad (1)$$

χ is distributed approximately as χ2 with 1 degree of freedom. For a 95% confidence test, $\chi^2_{1,095}$ = 3.84. So if χ is larger than 3.84, then with 95% confidence, the null hypothesis is rejected. In other words, the two classifiers have the same error rate. From the above data, the McNemar's test statistic has the value of 8.2045 with 1 degree of freedom. The two-tailed P value equals 0.0042, using conventional criteria; the difference between classifiers' performance is considered to be statistically significant. Hence, the null hypothesis is rejected, and it is accepted that the two classifiers have significantly different performances.

The result of the McNemar's test show that the SVM classifier for Experiment A (word without stemming) produced significantly better results than the SVM classifier for Experiment B (light stemmer). This outcome contradicts previous studies such as (Brahimi et al., 2016) and (Abdulla et al., 2013), showing that light stemming gives better results than words without stemming for Arabic tweets. However, it is noted that the dataset used by these two studies included a mixture of MSA texts and tweets in potential local Jordanian dialect, which could explain the difference between both scenarios. The experimental results in this section highlight one of the key contributions of this thesis.

Stemming has commonly been used for text classification problems in various languages. Recent research in this area shows that the impact on the classification process has varied, and it was therefore important to investigate the case for colloquial Arabic. For

instance, on considering the Indonesian language, Hidayatullah et al. (Hidayatullah et al., 2016) performed an experiment with tweets, and the results indicate that stemming does not consistently contribute to the accuracy for the SVM and the NB classifiers. In contrast, Basnur and Sensuse (Basnur and Sensuse, 2010) used stemming with the NB classifier to categorise news articles written in Indonesian, and it was found that stemming helped to enhance the accuracy of this process; clearly the text in this scenario is long compared to tweets. However, experiments by Torunoğlu et al. (Torunoğlu et al., 2011) and Can et al. (Can et al., 2008), which utilised the SVM, NB and k-NN classifiers, show conflicting results regarding the influence of stemming on Turkish text. Furthermore, in English, Aiello et al. (Aiello et al., 2013) have compared six topic detection methods on three Twitter datasets; their results show that the stemming phase reduces the performance of all examined methods. They refer to the negative effect of word stemming concerning the fact that stemming partially disrupts word associations by merging too many words together. Another project by Zangerle and Specht (Zangerle and Specht, 2014) investigated how Twitter users react to having their account hacked and how they deal with compromised accounts; in their experiment it was reported that word stemming did not increase performance for SVM. Nonetheless, Dovgopol and Nohelty (Dovgopol and Nohelty, 2015) found that stemming has a negligible impact on the performance of the NB and k-NN classifiers, and as a result of that, they excluded the stemming phase from their final algorithm. Similar to the results for colloquial Arabic in this section, the findings from experiments conducted on short English texts (mainly tweets) agree that word stemming does not improve the performance of the classifiers mentioned when applied to informal text.

Furthermore, the Arabic script has numerous diacritics such as consonant pointing and vowel marks, which are supplementary discitis. They are used to facilitate pronunciation or to

distinguish between a word and one of its homographs. In this study, a diacritics remover was applied to normalise words for two reasons: Firstly, diacritics are very rarely used (if any) in colloquial Arabic, for example (Habash, 2010) states that 98% of Arabic text is written without diacritics; secondly, because similar findings were observed in our own corpus.

## 5.5 Chapter summary

In addition to investigating the effect of light stemming on colloquial Arabic text for text classification, the main contribution of this chapter is to investigate a variety of text classification techniques using colloquial Arabic text as a dataset. The classification techniques used in this work have been widely used by many researchers to classify MSA text. However, to the best of our knowledge, none of the previous studies have tried to compare the performance of all these techniques when applied to datasets populated with colloquial Arabic text, as presented in this chapter. The classification algorithms tested in this study are: SVM, J48, C5.0, NNET, NB and k-NN. SVM produced the most accurate results. The next most noteworthy classification algorithms were Decision Tree (C5.0 and J48). For feature selection, experiments were carried out under a number of conditions: without stemming, with light stemming, with removing common prefixes in the Arabic language, and finally with removing common suffixes. The research findings suggest that most classifiers perform better without applying stemming. After detecting the flood event, the next step is to infer or extract the flood's location; therefore the next chapter will discuss how to infer location NE from tweets.

# Chapter 6: Location Inference from Twitter

## 6.1 Introduction

The ability to infer locations from social media platforms comes with immense benefits to service providers and consumers themselves. Twitter users' locations are important for many purposes, such as targeted advertising (Moshfeghi, 2016) and cyber bullying(Bellmore et al., 2015). However, for specific event detection purposes such as natural disasters, the event location which is directly mentioned in the tweet is more important and more accurate than user location to detect event location. This chapter investigates how to infer the event location of a specific type of high-risk natural disaster, namely 'floods' by applying an Arabic Named Entity Recognition (NER) technique, especially with informal (colloquial) Arabic text, which is commonly used on social media and has received very limited attention.

The remainder of this chapter sets out the problems and challenges and a list of spatial features that can be found in tweet content and metadata. Then, a Learning to Search (L2S) method will be discussed and it will be explained how to capitalise on it to develop the proposed location NER. After that, the applied experiments will be presented, including discussing the data collection stage, the pre-processing and classification stage, and location NER technique. Finally, the experimental and comparison results will be shared and discussed.

## 6.2 Problems and Challenges

This work proposes a location estimate method based on Arabic NER on colloquial Arabic text collected from Twitter. Locations such as country, city, town, or village are defined as ***geographic location*****,** and ***point-of interest (POI)*** refers to locations such as shopping centres and restaurants. In addition to the tweets' brevity (maximum 280 characters), the main challenges of dealing with colloquial Arabic tweets are that they are characterised as being highly ill-structured, inconsistent, and difficult to process. NLP tools therefore need to be adapted to work effectively on such text. This work also intends to test the following hypotheses:

The null hypothesis ($H_0$): "NER techniques cannot be applied to accurately estimate flood location NE in colloquial Arabic text".

The alternative hypothesis ($H_1$): "The flood location can be accurately estimated by recognising Arabic named entities, especially locations and organisations, which are mention in tweets".

The hypotheses will be tested by developing a method to infer location NE from colloquial Arabic tweets based on L2S, and comparing the proposed method's results with existing NER systems.

The assumptions used for the design of this work are:

***Assumption 1*** tweets that mention high-risk natural disasters contain the event location name in their content.

***Assumption 2*** the current location of the user (geotagging location) is identical to the location of the event mentioned in the tweet.

An Arabic NER technique was applied to extract location names for geotagged tweets, before examining the accuracy of the inferred location compared to the geotagging location information.

## 6.3 Types of Locations and Spatial Features on Twitter

Tweet content and metadata contain different types of location indicators, such as place names appearing in the message, a location from which the message was sent, and so on. In order to successfully infer event location from these indicators, it is necessary to consider which location indicators are appropriate for this research. For this reason, each type of location indicator has been defined and its relevance to this work presented.

***User profile location*** is a free-text format that enables users to manually type in their location or any other text. (Ikawa et al., 2013) found that Twitter users disclose their location at the level of a country or city, however most users do not report their home location.

***User profile time zone*** is the time zone the user declares themselves within at the level of country.

***User profile URL*** refers to the user's website or personal web page, which provides both the website's country code and the website's geocoded IP address. However, it is not accurate for location, as there is the possibility of hosting their website in one geographical location and living elsewhere (Atefeh and Khreich, 2015).

***Tweet geotag location***, geotagged data comes with digital latitude and longitude coordinates which are collected from GPS receivers and/or Wi-Fi. However, less than 0.5% share their geotagged location (Li et al., 2012).

***Locations in tweet context*** means location entities mentioned in Twitter messages. There are two type of ambiguities, geo/non-geo, which occur when location entity is identical to a

proper name (e.g. Baker is a given name and the name of a city in Montana, United States). On the other hand, geo/geo ambiguity happens when more than one place has the same name entity but are actually absurdly different, such as Moscow, Russia; Moscow, Kansas, USA; Waterloo, Belgium; Waterloo, Canada; Sydney, Australia; Sydney, Canada (Inkpen et al., 2015).

***User friends and followers networks*** could help in extracting the user location by utilising friends and followers' location data.

## 6.4 Methodology

The key phases used to infer the location NE of high-risk floods from Arabic tweets are illustrated in Figure 6-1. In this experiment, the methodology applied in section 5.3 was followed to collect Arabic tweets using R tools and to classify tweets. In addition to collecting Arabic tweets that mention floods or torrents ("فيضانات,سيول "), the focus was on geotagged tweets to compare the performance and results of the actual location to the inferred one, and verify the quality of the estimation. The idea of the proposed method is that the Twitter user who mentions high-risk natural disasters will also mention the location of the event. The method has three stages:

1. Data gathering stage; this stage will discuss how the data set was collected.
2. Pre-processing and classification stage shows the steps applied to filter the tweets collected and predict the associated class based on a pre-trained model.
3. Infer the location NE from tweet content by utilising the proposed location NER, which is based on the L2S method.

Although there are several data mining tools available, such as Weka, RapidMiner, and Orange, R and python languages were selected to analyse the data. In this experiment, R and Python were integrated using *R Markdown* (Porte, 2017), which allowed us to transfer data between R and Python.

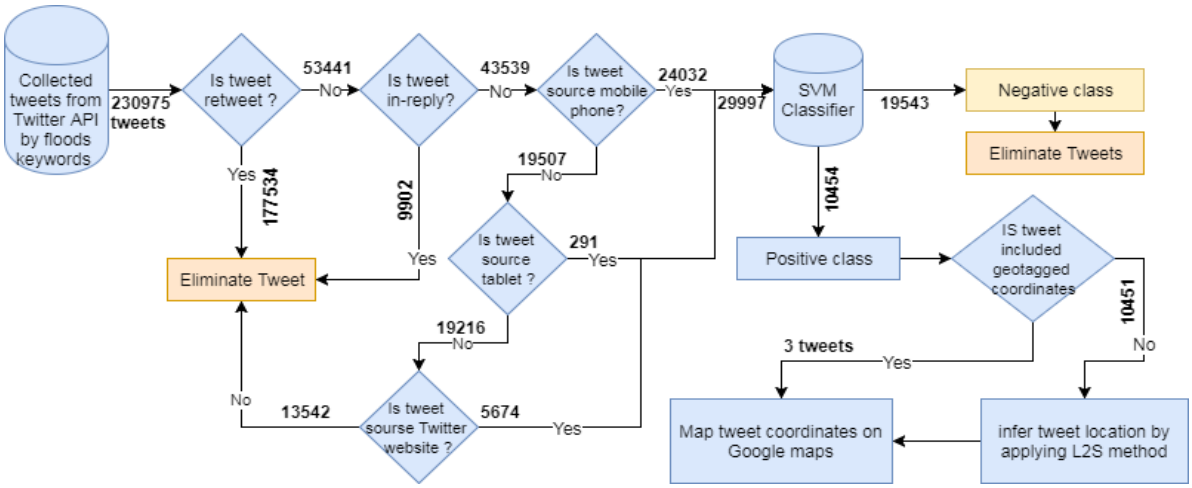


*Figure 6-1 Main steps to infer high risk floods location from tweets*

### 6.4.1 Tweet gathering stage

Twitter Streaming API (Twitter, 2018a) were accessed using the TwitterR package to collect flood data. Twitter API functions allow the filtering of tweets based on their language; the function is utilised to collect tweets written in the Arabic language. Only tweets directly mentioning 'floods' or 'torrents' ('سيول OR فيضانات') in Arabic were collected, and then the crawler was run from 05 May 2017 to 01 Jun 2017, and eventually a total of 230975 unique tweets were collected. As the experiment needed to calculate the distance value between the geotagged location and the inferred location, tweets with geotagged locations were collected to evaluate the accuracy of the proposed location inferring method. The total number of tweets with geotagged location is 41 tweets, which is less than 0.02% of the tweets collected. Most of those geotagged tweets are automated tweets which were sent from

applications linked with a Twitter account. In this case, those automated tweets have been considered as spam tweets, and for that reason only tweets sent from mobile phones, tablets or trusted official websites (e.g. Twitter, Facebook and Google) were included in analysis stage. Furthermore, retweeted and reply tweets were excluded from the analysis stage to focus on real time tweets. The number of tweets that passed to the pre-processing stage is 29997 tweets.

Figure 6-2 shows the number of tweets per class and tweet source. It shows that most of the tweets collected came from smart phones (80%), which includes iPhone (48%), Android (32%), BlackBerry (0.04%) and Windows (0.036%).

### 6.4.2 Text pre-processing

In this stage, pre-processing steps were applied, which are discussed in section 5.3.3. These include:

- Remove punctuation marks and numbers
- Remove all non-Arabic characters
- Remove diacritics
- Letter normalisation
- Exclude stop words list
- Apply light stemming
- Generate TF-IDF matrix

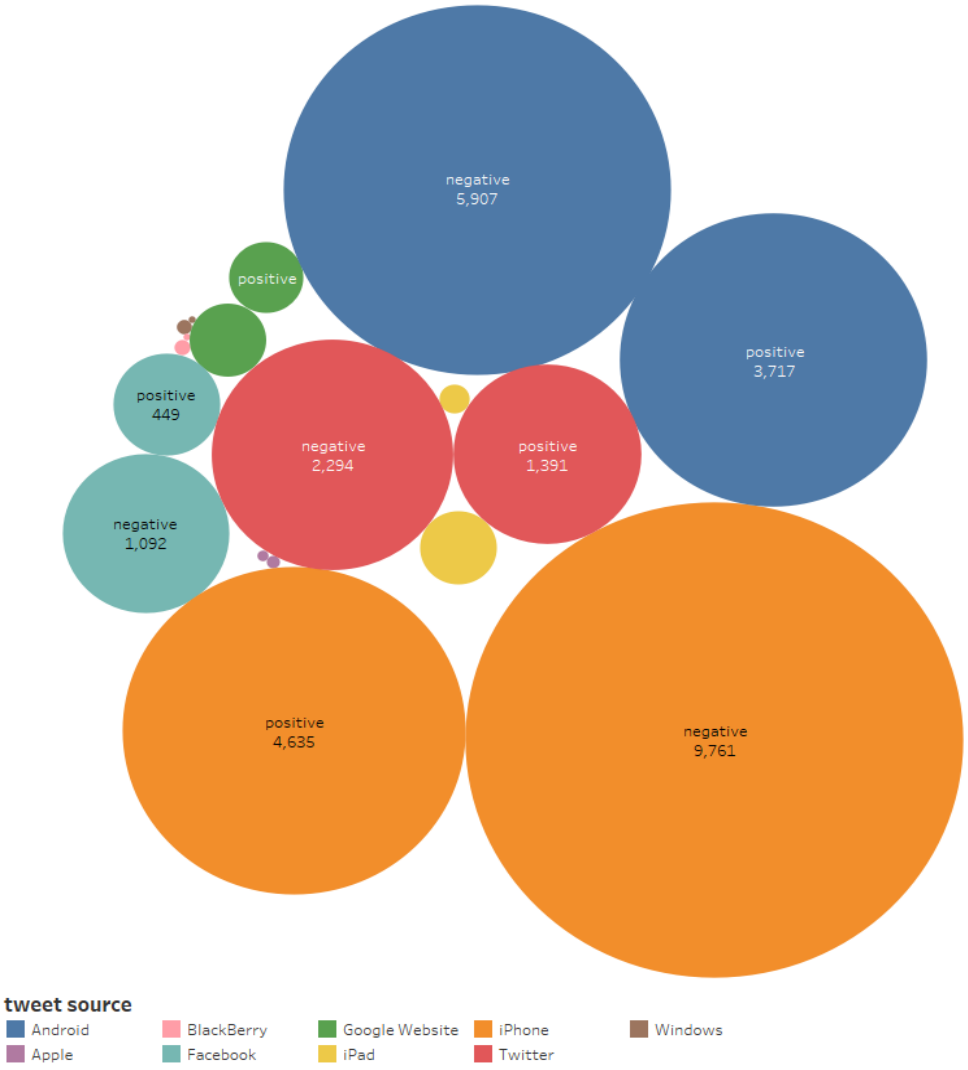


*Figure 6-2 Number of tweets by source*

The output of this step is the TF-IDF matrix, which contains 29997 lines. Each line represents single filtered tweets, and the columns represent the Arabic words that appear in those tweets.

### 6.4.3 Classification stage

In this stage, the collected tweets were classified as positive class or negative class; the positive class contains tweets mentioning high-risk floods, and the negative class contains the

remaining tweets in the dataset. To classify the tweets collected, a trained support vector machine (SVM) classifier was applied, which was produced as set out in section 5.3.6.

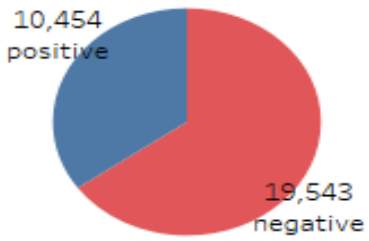


*Figure 6-3 Number of tweets per class*

Figure 6-3 shows that 35% of the tweets mention high risk floods, but nevertheless less than 0.03% of the tweets in the positive class include geotagging location information, which led to providing strong evidence that geotagged location is not enough to infer event location as a consequence of the small number of geotagged tweets. As a result, tweet location has been inferred from the tweet context to detect the event location.

The output of this stage is 10454 filtered tweets that relate to high risk floods, and those tweets passed on to the next step to extract the location mentioned by using the L2S approach.

### 6.4.4 Location Named Entity Recognition

As mentioned in section 3.4.3, the aim of this research is to infer event location NE and ignore another location NEs mentioned in tweets. The state of the art NER Systems based on gazetteers or conditional Random Field (CRF) models underperformed when tested on the Twitter corpus for both the Arabic and English languages (Zirikly and Diab, 2015); (Derczynski et al., 2015), and such systems do not distinguish the event location NE from other location

NEs which are mentioned in a tweet. To solve this issue, a location NER method has been developed based on the learning-to-search (L2S) approach (Daumé III et al., 2014).

L2S is one approach to solving structured prediction tasks. It is the minimal complexity approach to producing a structured prediction. This approach has been used to solve joint predictions in natural language processing (NLP) problems such as NER and part of speech (POS).

The results of (Daumé III et al., 2014) show that L2S approaches have been demonstrated to be competitive with other structured prediction approaches and CRF models, both in accuracy and running time. Moreover, they state that existing L2S algorithms guarantee that if the learning step performs well, then the learned policy is almost as good as the reference policy, implicitly assuming that the reference policy achieves good performance. Good reference policies are typically derived using labels in the training data, such as assigning each word to its correct POS tag.

A named entity annotator was developed using a BIO chunking classifier, which labels each token as the Beginning (B), the Inside (I) or entirely Outside (O) the span of a named entity. As this task's target is to infer event location, three label type sets of entity tags were applied, included Location (LOC), City (CIT) and preposition (PRE). Initial observations suggest that each tweet reporting a high risk flood mentioned event location as a point of interest (POI) location and the city, town, or village. For that reason, POI locations and city locations were separated to link these locations together to obtain better accuracy when mapping it onto Google maps. Preposition entity (PRE) refers to common terms that appear with location entities and help the proposed method to detect locations entities; the example sentences in this encoding are shown in Figure 6-4.

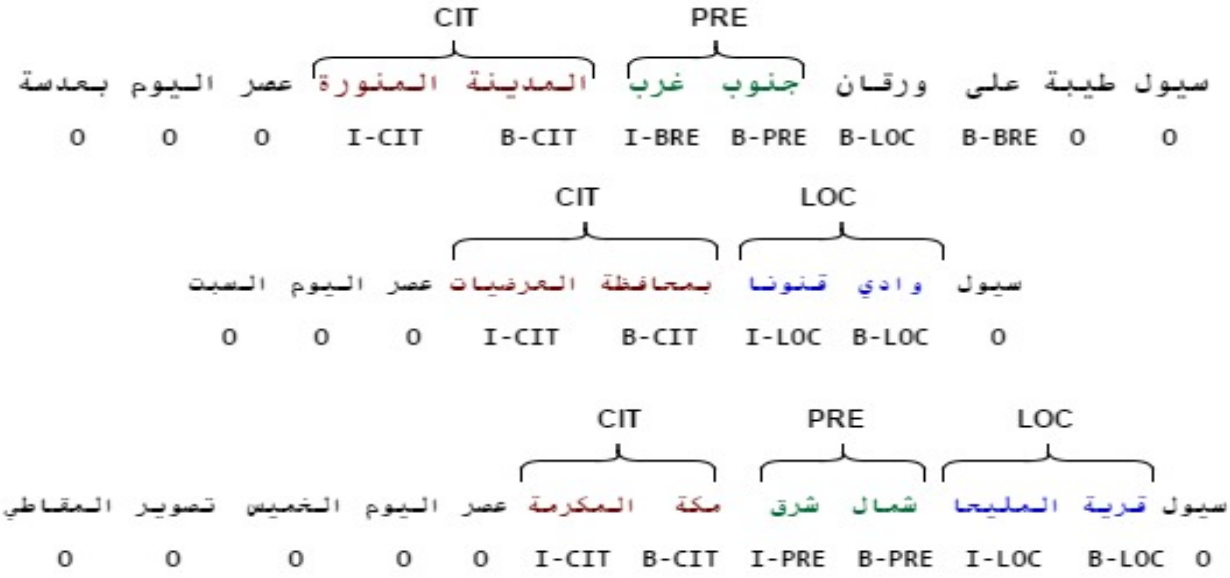


*Figure 6-4 Example inputs and desired outputs for named entity recognition task.*

Each word in 300 tweets was tagged using a BIO tag scheme. 70% of those tweets have been utilised for training purposes, and the rest of the tweets for testing a trained model to evaluate the accuracy of the location inferring approach.

## 6.5 Results, Analysis and Discussion

The proposed location NER method solely identifies event location and omits other locations mentioned in tweets. For the evaluation, the tweets in the test dataset were utilised to compare the proposed method with existing Arabic named entity recognition systems. The first system is FARASA (Institute, 2017), which is based on work reported by (Darwish and Gao, 2014); their system was adopted since it produces near state-of-the-art results. The second system adopted is Polyglot-NER (Al-Rfou, 2017), which is the work proposed by (Al-Rfou et al., 2015). Figure 6-5 shows an example of results when tested with a test dataset, the highlights in the blue colour and underlined (one line) represent a correct event location NE,

and the highlights in red colour and underlined (two lines) represent an incorrect event location NE.

***Example 1:***

Filtered tweet: "سيول شرق تربة البقوم شرق الطائف اليوم الاحد ه رصد".

FARASA: "سيول شرق تربة البقوم شرق الطائف اليوم الاحد ه رصد".

Polyglot-NER: "سيول شرق تربة البقوم شرق الطائف اليوم الاحد ه رصد".

Our system: "سيول شرق تربة البقوم شرق الطائف اليوم الاحد ه رصد".

***Example 2:***

Filtered tweet: "سيول قرية هبت جنوب غرب المدينة الان عضو الفريق المبدع عمار الاحمدي مكشات المدينة طقس".

FARASA: "سيول قرية هبت جنوب غرب المدينة الان عضو الفريق المبدع عمار الاحمدي مكشات المدينة طقس".

Polyglot-NER: "سيول قرية هبت جنوب غرب المدينة الان عضو الفريق المبدع عمار الاحمدي مكشات المدينة طقس".

Our system: "سيول قرية هبت جنوب غرب المدينة الان عضو الفريق المبدع عمار الاحمدي مكشات المدينة طقس".

***Example 3:***

Filtered tweet: "بالصور امطار وسيول شرق القراصه شرق العيص الثلاثاء ".

FARASA: "بالصور امطار وسيول شرق القراصه شرق العيص الثلاثاء".

Polyglot-NER: "بالصور امطار وسيول شرق القراصه شرق العيص الثلاثاء".

Our system: "بالصور امطار وسيول شرق القراصه شرق العيص الثلاثاء".

*Figure 6-5 Examples of location NER systems results. It denote error as the following true positive , false positive and false negative*

Recall, precision, and F-measure are usually used to measure the system's performance in location NER. Recall, precision and F-measure were computed as follows:

$$\text{Precision} = \frac{\text{number of correct NE recognized by the system (TP)}}{\text{number of NE givenby the system (TP+FP)}} \quad (2)$$

$$Recall = \frac{\text{number of correct NE recognized by the system (TP)}}{\text{number of correct NE in the corpus (TP+TN)}} \quad (3)$$

$$\text{F measure} = \frac{2*(\text{precision}*\text{recall})}{(\text{precision}+\text{recall})} \quad (4)$$

Where TP is True Positive, FP is False Positive, and TN is True Negative.

The comparative results of the NER systems in Table 6-1 highlight one of the key contributions of this thesis. The results from testing data represent the accuracy of locations by the NER systems, specifically, the ability to recognise the location from colloquial Arabic text obtained from Twitter, are shown in Table 6-1. This table shows which technologies are best at detecting locations using colloquial Arabic text. Table 6-1 shows the F1, recall and precision results for 100 tweets in a test dataset generated by the three location NER systems (see Appendix A "Test data of NER task" for details of NER systems performance). As shown in Table 6-1 the proposed system outperformed the other systems and achieved a degree for F1 of 86%. After analysing Table 6-1 which shows that the recall results, it has been demonstrated that the FARASA system achieved the worst performance. The main reason for that is because the FARASA system is based on cross-lingual links between Arabic and English, which means the FARASA system translates Arabic text into English in order to exploit discriminative features and large resources in the English language. As mentioned by (Darwish and Gao, 2014), the FARASA system's accuracy performance dropped significantly when tested with the Twitter dataset, and some of the factors that were observed are:

1) some words that would typically be regular words are recognised as location NE, for example " سيول " which mean "floods" are recognised as location NE referenced to Seoul "سيئول - سيول" the capital city of South Korea).

2) Some location NEs in the tweet dataset are unknown or not common, and when translated into English the translator deals with it as a regular word.

The Polypglot-NER system achieved better accuracy in the results compared to the FARASA system, however, this result is still far away from the proposed system's results. Polypglot-NER is based on a word level classification problem; it utilises Wikipedia to create a named entity training corpus. The main reason why the proposed system outperformed the Polypglot-NER system is that the Polypglot-NER system aims to recognise all location NE mentioned in tweets, whereas the proposed system aims to recognise event location NE and ignore other location NEs mentioned in the tweet. Moreover, the proposed system considers the neighbouring words and the word order in the sentence (tweet) to distinguish the event location NE and disregard other words.

*Table 6-1 Location NER systems results*

| | **precision** | **Recall** | **F1** |
|---|---|---|---|
| **FARASA** | **60.39** | **23.01** | **33.32** |
| **Polyglot-NER** | **62.68** | **47.90** | **54.30** |
| **Proposed system** | **96.36** | **77.94** | **86.17** |

Table 6-2 presents the performance of each NER system with three example tweets (the tweets are presented in figure 6-5). For example, the tweet " سيول شرق تربة البقوم شرق الطائف اليوم الاحد ه رصد" is made up of 10 words, and the proposed system returned three words as the location NE: "تربة البقوم الطائف". Figure 6-5 shows that the location NE's for this tweet are " تربة " and " الطائف ", therefore the TP = 2 . However, the proposed system also provided the location NE "البقوم ", but this is incorrect, therefore the FP =1. FN represents the number of words that are NEs, yet that the system did not able to recognise, for this example the FN =0. Any remaining words represent TN, which are not location NEs, and the proposed system

recognised it correctly (TN = 7). The precision, recall and F1 values for three location NE systems have been calculated according to the TP, FP, FN and TN values.

*Table 6-2 NER systems perform on 3 example tweets (TP = True Positive, FP= False Positive, FN= False Negative, TN= True Negative,)*

| Tweet ID | Inferred Location NE by proposed System | Proposed NER system | | | | Farasa System | | | | Polyglot-NER System | | | | words in tweet |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | TP | FP | FN | TN | TP | FP | FN | TN | TP | FP | FN | TN | |
| 1 | تربة البقوم الطائف | 2 | 1 | 0 | 7 | 0 | 1 | 2 | 7 | 2 | 0 | 2 | 6 | 10 |
| 3 | قرية هبت المدينة | 3 | 0 | 0 | 12 | 0 | 0 | 3 | 12 | 0 | 3 | 1 | 11 | 15 |
| 67 | القراصه العيص | 2 | 0 | 0 | 7 | 0 | 0 | 2 | 7 | 1 | 0 | 1 | 7 | 9 |

## 6.6 Chapter summary

This chapter has identified the main research challenges in inferring the location NE for high-risk floods from Arabic tweets, and has formatted the research hypotheses which have been tested in this chapter. Subsequently, the flood location NER approach has been developed using the L2S method, along with developing a named entity annotator using a BIO tag scheme. The flood location NER task was tackled by producing a comparative experiment to compare and evaluate the performance of the proposed system with FARASA and Polyglot-NER systems. The experiments' results show that the proposed system significantly outperformed other systems when applied to colloquial Arabic text collected from Twitter. Hence, the null hypotheses have been rejected and it is accepted that the flood location can be accurately estimated by recognising Arabic named entities, especially locations and organisations which are mentioned in the tweets. The next chapter discusses and addresses location NEL problems, and will use the location NE identified using the proposed NER system.

# Chapter 7: Locations Named Entity Linking

## 7.1 Introduction

In Natural Language Processing (NLP), the named-entity mentioned may refer to several entities, and the process of determining the appropriate meaning in context is called Named Entity Linking (NEL) or named entity disambiguation (NED). NEL is the task of linking a named entity that is mentioned in the text to an instance in a knowledge base (Derczynski et al., 2015). For example, the sentence “Ronaldo is a Portuguese professional footballer who plays as a forward for Real based in Madrid.” The natural language API understands that “Ronaldo” refers to Cristiano Ronaldo, that “Portuguese" is his nationality, that "Real" refers to Real Madrid football club, and finally, that “Madrid” refers to the city of Madrid in Spain. The task is not trivial, as multiple named entities can be referred to by the same name. The location name “Madrid”, for instance, is associated with multiple entities like the capital city of Spain, or a town in the state of New York, USA, or a town in the state of Alabama, USA, among many others. Given a piece of text, the Google API (Google, 2018a) will link all location NEs to a geocode based knowledge graph.

This chapter presents the proposed method to link and map location NEs extracted from tweets to maps using a freely available geocoding service provided by Google. The most well-known free geocoding services are Google and OpenStreetMap (OSM). The geocoding API service from Google has been applied in this work dependent on recent comparative research (Lemke et al., 2015), which compared between Google and OSM geocoding services in terms of matching rate and positional accuracy. (Lemke et al., 2015) data set consists of 2500 randomly unique addresses, the results state that Google geocoding services

outperformed the OSM geocoding service regarding both completeness (Google > 93% vs OSM > 82%) and positional accuracy (the addresses were geocoded within <50 meters), where the Google service achieved an accuracy of nearly 95%, and 50% for the OSM service.

The remainder of this chapter deals with the problems and challenges, and explains the proposed location NEL method, as well as how to use Google maps API. Finally, there will be a discussion of the proposed location NEL and the results.

## 7.2 Problems and Challenges

Microblog NEL is a relatively new, underexplored task. Most of the studies of location NEL in Twitter have used Wikipedia as the knowledge base (Geiß et al., 2017, Li et al., 2015, Liu et al., 2013, Zirikly and Diab, 2015). The location NEL techniques link tweets to Wikipedia pages, which typically represent entities and concepts.

In our case, to determine and link the location mentioned in colloquial Arabic tweets, a knowledge base in location NE was needed. The aim of applying an entity link in this work was to identify the most accurate location associated with the location NE in tweets. The challenges of location NEL in colloquial Arabic tweets can be listed as:

- Twitter users often use a brief location NE
- Twitter users sometimes use a locally know location NE, which may be difficult to associate with the knowledge base
- Some location NEs in Arabic refer to multiple places
- Ambiguous geo-locations (returned more than one geolocations) or foreign geo-locations

The proposed NEL method aims to address those challenges by utilising Google maps API services. Accuracy within specified ranges (10, 20, 30, 40, 50 and 100 km), error distance and average error distance, have been reported to validate the effectiveness of the proposed method.

## 7.3 Location NEL Method

Google provides several maps API services, including but not limited to, directions, geocoding, geolocation, roads and places. A geocoding task is the process of converting textual addresses into associated latitude (lat) and longitude (long) coordinates, for example "Eiffel Tower, Paris" associated with (lat: 48.8584° N, long: 2.2945° E), which helps to point out the addresses on the map. Google provides a free geocoding API service which is restricted to a limit of allowed requests for 2500 addresses per 24 hours, and to avoid this limit, Google also provides a premium service (Google, 2018b).

This section will show the proposed method to link the flood location, as inferred in Chapter6. Algorithm 3 shows the implementation pseudo-code of the proposed method to link and map each tweet to google maps. As shown in section 6.4, each inferred location NE in tweets was tagged as (LOC) or (CIT) where LOC refers to POI locations and CIT refers to city, town, or village locations. The input of Algorithm 3 will be a set of filtered tweets ($\boldsymbol{A}$); each tweet consists of pairs of (***x***, ***y***), where ***x*** indicates the Arabic words mentioned in the filtered tweet, and ***y*** indicate the ***BIO*** tag associated with ***x***. The output of Algorithm 3 will be the geocode coordinate (***lat***, ***long***) for each tweet belonging to $\boldsymbol{A}$. The first step in the Algorithm 3 process for each filtered tweet in $\boldsymbol{A}$ a set $\boldsymbol{t} \subseteq \boldsymbol{A}$ which consists of each ***x*** tagged as "B-LOC",

"I-LOC", "B-CIT" or "I- CIT", and those entity tags refer to the location NEs, as discussed in Chapter 6:. Assuming that there is no word tagged as the location NE in the tweet, Algorithm 3 will return an NA (not available) value for the tweet geocode coordinate; otherwise, it will send a request to the Google geocode API to search for geocode coordinates associated with the textual address, which includes all location NEs in the $tweet_i$. Supposing that Google returns the geocode coordinates ($lat_0$, $long_0$) for the submitted request, then ($lat_0$, $long_0$) will be located as an inferred location for the $tweet_i$, otherwise, in the case that the Google geocode API returns zero results for the geocode coordinate, then $\boldsymbol{t}$ is divided into two sets $t_1$ and $t_2$, $t_1$ is included for each $\boldsymbol{x}$ tagged as "B-LOC" or "I-LOC" and $t_2$ is included for each $\boldsymbol{x}$ tagged as "B-CIT" or "I- CIT". After dividing the $\boldsymbol{t}$ **in**to two, requests are submitted to the Google geocode API. Each request consists of a $t_1$ and $t_2$ location NE, which will return one of four probable cases:

1) Google geocode API returns zero results for the geocode coordinate for both $t_1$ and $t_2$ inquiries, so Algorithm 3 will implicitly assume that the $t_i$ geocode coordinate is NA (not available) and then attempt to extract a $t_i$ geocode coordinate value by applying a while loop to exclude the last word in $t_i$ until Google API returns a geocode coordinate or the loop is ended.

2) Google geocode API returns zero results for the geocode coordinate for $t_2$ inquiry and returns ($lat_1$, $long_1$) for the $t_1$ inquiry, so Algorithm 3 will locate ($lat_1$, $long_1$) an inferred location for the $tweet_i$.

3) Google geocode API returns a zero result for the geocode coordinate for the $t_1$ inquiry and returns ($lat_2$, $long_2$) for the $t_2$ inquiry, so Algorithm 3 will locate ($lat_2$, $long_2$) as the inferred location for the $tweet_i$.

4) Google geocode API returns ($lat_1$, $long_1$) for the $t_1$ inquiry and returns ($lat_2$, $long_2$) for the $t_2$ inquiry, so Algorithm 3 will utilise the "Haversine" formula to calculate the distance (***d)*** between ($lat_1$, $lon_1$) and ($lat_2$ , $lon_2$). In the case that $d \geq 50$ km, this gives a very good indication of ($lat_1$, $lon_1$), which refers to the POI location being located far away from ($lat_2$, $lon_2$), which refers to a city, village or town. In this case, it can be implicitly assumed that the Google geocoding API has returned a wrong POI location ($lat_1$, $lon_1$) for another city as a result of the similarity of POI location names. POI locations include, but are not limited to, residential neighbourhoods, schools, town halls, buildings or tourist attractions, which may have a similar name to this location in another city. For that reason ($lat_2$, $lon_2$) were assigned as the inferred locations for $tweet_i$ in this case. On the other hand, in the case that $d < 50$ km, Algorithm 3 will locate ($lat_1$, $long_1$) as the inferred location for the $tweet_i$. In this case, it is implicitly assumed that Google geocoding API returns a correct POI location ($lat_1$, $lon_1$) and POI location is more accurate than city location.

During the process of designing Algorithm 3, the following points were considered:

- Misspellings: location NEs from twitter where typing errors and misspellings are common, and these errors may occur for location; as a result of this error, the knowledge bases (e.g. Google, OSM and Wikipedia) probably are not able to link this location NEs to geocode coordinates. To solve this problem, Algorithm 3 attempts to avoid the location NEs with typing errors by removing them from the knowledge base search inquire.

- Mis-tagging: the location NEL method analysis and process location NEs that flow from the location NER; as discussed in Section 6.5, the NER systems may

tag non-location entities as a location entity, which will have an effect on NEL performance.

- Data completeness: Algorithm 3 aims to associate each tweet with geocode coordinates and avoid the NA value as much as possible.
- Positional accuracy: the main aim of the location NEL method is to link the location NE in tweets with the most accurate geocode coordinates.

**Algorithm 3**

***Input***: $A$ = {tweet$_1$,tweet$_2$,…,tweet$_i$,...,tweet$_n$} ; each tweets as (word, tag) pairs.

tweet$_i$ = {(word$_1$,tag$_1$),(word$_2$,tag$_2$),….,(word$_j$,tag$_j$),…..,(word$_m$,tag$_m$)}

***Output***: geocode coordinate for each tweet ∈ $A$

1 ***for*** ∀ tweet$_i$ ∈ $A$ ***do***

2 t ={(word, tag) : words tagged as “B-LOC” OR “I-LOC” OR “B-CIT” OR “I- CIT”}.

3 ***if*** $t \neq \emptyset$ ***then***

3 send google geocode inquire with text = “word$_1$ word$_2$ …word$_j$ …word$_n$” : word$_i$ ∈ $t$

4 ***if*** google ***return*** geocode coordinate (lat$_0$,lon$_0$) ***then***

5 Tweet$_i$ geocode coordinate = (lat$_0$ ,lon$_0$)

6 ***else***

7 divide t to tow sets

8 t$_1$ ={(word, tag) : words tagged as “B-LOC” OR “I-LOC”}.

9 t$_2$ ={(word, tag) : words tagged as “B-CIT” OR “I- CIT”}.

10 send google geocode inquire with text =”word$_1$ word$_2$ …word$_j$ …word$_n$” : word$_i$ ∈ t$_1$

```
11          if google return geocode coordinate (lat1 ,lon1) then
12           send google geocode inquire with   text ="word1 …wordj …wordm" : wordi ∈ t2
13             if google return geocode coordinate (lat2 ,lon2) then
14                apply "Haversine" to calculate d distance between (lat1 ,lon1) and (lat2  ,lon2)
15                  if d ≥ 50 km then  return  Tweeti geocode coordinate = (lat2 ,lon2)
16                 else  return  Tweeti geocode coordinate = (lat1 ,lon1)
17             else return  Tweeti geocode coordinate = (lat2 ,lon2)
18          else  send google geocode inquire with   text ="word1 …wordj …wordm" : wordi ∈ t2
19              if google return geocode coordinate (lat3 ,lon3) then
20                 return  Tweeti geocode coordinate = (lat3 ,lon3)
21              else  k = length t- 1   (t length is the number of words in t)
22                      Tweeti geocode coordinate = NA
23                      while (k >= 1)
24                            t4 = { word1, word2, …, wordn,…,wordk}
25                            send google geocode inquire with t4
26                           if google return  geocode coordinate (lat4 ,lon4) then
27                              Tweeti geocode coordinate = (lat4,lon4)
28                              break while loop
29                           else k = k-1
30                       end while loop
31                       return Tweeti geocode coordinate
32  end for
```

For Algorithm 3, the inputs are the location NE, tagged as (LOC) or (CIT). In addition, Google geocode API has also been used to return the latitude and longitude coordinates for a variety of address types as outputs. The most important address types returned by Google geocode API (Google, 2018, Google Maps Geocoding API) are defined in Table 7-1 There are differences in the number of inputs and outputs for some address types (2 inputs types, 21 outputs types), which could result in the misclassification of the location type; moreover, this may reduce the

accuracy in the performance of the proposed system, especially during the NEL stage. This misclassification may occur due to google API classifying POI locations (tagged as LOC) as cities, towns or villages, and this may also occur the other way around. Therefore, the structure of Algorithm 3 has been developed and built to address such misclassifications through a process of distinguishing between locations tagged as LOC or CIT, and connecting the data with the most appropriate location type in google API according to what is found when the Google geocode API returns the latitude and longitude coordinates for two different locations. This is enabled by Algorithm 3 calculating the "Haversine" formula in order to discover the distance between the locations returned. Next, Algorithm 3 provides the latitude and longitude coordinates for a single location based on the distance value between these two locations and the distance threshold (50 km). Abdelatti (2017) calculated the distance threshold in their study on the nature and trend of urban growth in Saudi Arabia, and they describe the urban area size of Hofuf city as being 50.41 $km^2$. While Hofuf is the largest city in Al-Ahsa Province, it is classified as medium size when compared to other cities in Saudi Arabia. However, The distance threshold could be changed to generalize for other countries, the threshold value depends on the country size and their cities size.

*Table 7-1 address types that returns by Google geocode API (Google, 2018a)*

| Type of Address | Description Indicators |
|---|---|
| Street Address | The specific street address |
| Route | A named route (e.g. Riyadh 1) |
| Political aspects | A specific political entity, often a polygon representing some sort of civil administration |
| Specific country | A national political entity; usually the highest order type returned by the Geocoder |
| Administrative_area_level_1 | A first-order civil entity below the country level. In the US, for example, these administrative levels refer to states, but not all countries have these administrative levels. Usually, for administrative_area_level_1, short names will match ISO 3166-2 subdivisions, as well as other popular lists; although, this cannot be |

| | |
|---|---|
| | guaranteed because the geocoding results are based on different signals and location data. |
| Administrative_area_level_2 | A second-order civil entity below country level. In the US, for example, these administrative levels are counties, but, again, not all countries have these administrative levels. |
| Administrative_area_level_3 | A third-order civil entity below the country level, likely a minor civil division, but not all countries have these administrative levels. |
| Administrative_area_level_4 | A fourth-order civil entity below the country level, usually a minor civil division. Not all countries have these administrative levels. |
| Administrative_area_level_5 | A fifth-order civil entity below country level, usually suggestive of a minor civil division, but not all countries have these administrative levels. |
| Colloquial name for an area | An alternative name commonly used for the entity. |
| Locality | A city or town political entity that is incorporated. |
| Neighbourhood | A specific named neighbourhood |
| Points of interest | A named point of interest, often prominent local entities that do not entirely fit into another category, for example the "Empire State Building" or the "Statue of Liberty." |

## 7.4 Location NEL Results and Discussion

As mentioned in section 6.4.1, most of the geotagged tweets are automated tweets which are considered to be spam tweets. Because of that, the latitude and longitude of 100 random tweets from the positive class were manually recorded to calculate the error rate for the proposed location inferring method (see Appendix B "Test data of NEL task" for details of inferred locations). Table 7-2 shows ten examples of tweets after the execution of the NEL method. The "Tweet ID" field is the unique identifier of a tweet assigned by Twitter, and "Inferred location name" field contains location name entities which are inferred from the tweet context. The inferred location fields show the inferred coordinates from Google Maps API, and the actual location refers to the manually input coordinates that refer to the correct location. The "Distance Error" field calculates the distance between the actual location and

the inferred location of a tweet, which is used as the evaluation metric to measure the accuracy of the results.

*Table 7-2 Example of the NEL method results to link and mapped tweets*

| No. | Tweet ID | Inferred location name | Inferred Location | | Actual Location | | Distance Error (KM) |
|---|---|---|---|---|---|---|---|
| | | | latitude | longitude | latitude | Longitude | |
| 1 | 863,758,320,418,639 | "تربة البقوم الطائف" | 21.21799 | 41.62535 | 21.2076 | 41.62114 | 1.235 |
| 2 | 863,754,673,416,794 | "قرية هبت المدينة" | 24.05006 | 39.19077 | 24.05006 | 39.19077 | 0 |
| 3 | 863,751,209,764,741 | "البيضا مكه" | 21.38546 | 39.87369 | 21.10848 | 39.91523 | 31.1 |
| 4 | 863,747,571,797,422 | "العيص تغلق" | 25.05994 | 38.11602 | 25.05994 | 38.11602 | 0 |
| 5 | 863,743,588,164,935 | "وادي حبونا" | -38.1556 | 145.9567 | 17.8411 | 44.02468 | 12250 |
| 6 | 863,716,042,673,205 | "وادي نجران" | 17.48086 | 44.20888 | 17.49178 | 44.15611 | 5.727 |
| 7 | 863,715,147,537,420 | "" | | | | | |
| 8 | 863,712,369,012,244 | "وادي فنونا العرضيات" | 19.44057 | 41.71891 | 19.44058 | 41.71891 | 0 |
| 9 | 863,696,656,977,494 | "وادي البرداني بارق" | 28.37623 | 45.95264 | 18.93053 | 41.92957 | 1127 |
| 10 | 863,649,596,513,423 | "محافظة العيص" | 25.05994 | 38.11602 | 25.17798 | 38.43767 | 34.94 |

The accuracy of the proposed method has been evaluated by calculating the distance between the inferred coordinates and the coordinates of the actual location of the tweets by using the “Haversine” formula. This formula calculates the great-circle distance as the shortest distance between two points based on the given coordinates. This method assumes a spherical earth, ignoring ellipsoidal effects. For instance, let us assume that there are two points as $P_1 = (\varphi_1, \lambda_1)$ and $P_2 = (\varphi_2, \lambda_2)$, then the distance between these two points can be calculated using the following equations:

$$a = \sin^2(\Delta\phi/2) + \cos\phi_1 \cdot \cos\phi_2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{(1-a)})$$

$$\text{Distance} = r \cdot c$$

Where r is the radius of the sphere, which is approximately equal to 6371 km.

The results achieved by the proposed NEL method shown in Table 7-3, which highlights one of the key contributions of this thesis. These results are based on 100 random tweets

from the data set. There were five tweets out of 100 returned with a non-location NE, and as a result of that, the NEL method will return an NA (not available) value for these tweets' geocode coordinates, as described in Algorithm 3. These five tweets have not been recorded in the results shown in Table 7-3 because there is no actual location for these tweets. As shown in Table 7-3, the location NEL method located 54 tweets within a 10 km distance from the actual location, and three tweets were located at distances between 10 and 20km from the actual location. On the other hand, there are 20 tweets that were located at distances greater than 50 km. In terms of the completeness of the geocoding results, out of 95 search inquiries (each inquiry represents a single tweet with location NEs), Google API returned NA (not available) for five tweets. According to this result, the proposed method has achieved a completeness degree of 94.7%.

*Table 7-3 Accuracy of the proposed NEL method*

| **Coverage radius with in (km)** | **10** | **20** | **30** | **40** | **50** | **100** |
|---|---|---|---|---|---|---|
| **Accuracy (%)** | **56.8** | **60** | **67.3** | **70.5** | **78.9** | **84.2** |

After an analysis of the proposed NEL method results, it was found that:

- The NA values are recorded when the location NER method tags non location NEs as LOG or CIT, then passes these non location NEs to Google API.
- The extracted geocode coordinates which are located at distances of between 10 and 50 km from the actual location refer to cities instead of POI locations inside those cities, because of several factors, including: (i) the NER method recognises cities' NEs and is not able to recognise POI location NEs mentioned in tweets, and as a result of that, only cities' NEs were passed to Google API. (ii) In cases of tweets

containing more than four inferred NEs, Google API faces difficulty in identifying such long inquires and returns NA values. As shown in section 7.3, the NEL methods divide NEs in those cases, as explained in Algorithm 3, and send inquires that return the geocode coordinates of cities.

- The extracted geocode coordinates, which are located at distances between 50 and 100 km from the actual location, refer to regions instead of POI locations inside those regions. These cases are reported when the actual location refers to rural areas.

To validate the effectiveness of the proposed location NEL method against other methods, it should be considered that the Arabic language faces additional issues when compared to Latin-based languages, such as the lack of annotated data sources or lack of comprehensive gazetteers.

Location NEL methods are often considered as the final step for location inferring tasks, and as a result of that, most studies in the literature have recorded the results for whole location inferring methods, as well as the proposed location NEL results, which are shown in Table 7-3; these are dependent on the NER method which was presented in Chapter 6. In the proposed location NEL, different data sets were used, different experiment conditions and different languages from studies in the literature. Hence, it is not feasible to statistically compare the proposed method's accuracy values with others. However, a survey on location inference techniques on Twitter (Ajao et al., 2015) states that the most recent work (Ryoo and Moon, 2014) on location NEL has achieved a 60% accuracy within 10 km; however, their work focuses on inferring the user's main location rather than tweet location. On the other hand, some studies (Ikawa et al., 2013, Schulz et al., 2013) have focused on location NEL,

which are inferred from tweets' content, and their work achieved 37% and 20% accuracy respectively within a 10 km coverage radius.

## 7.5 Chapter summary

In this chapter, the NEL tasks have been discussed in detail and the challenges that are faced when dealing with colloquial Arabic text have been listed. In summary, a NEL method was designed that aimed to utilise Google API services as a knowledge base to extract accurate geocode coordinates that are associated with the location NEs mentioned in tweets. Considering that misspellings occur in tweets and location NER mis-tagging, the location NEL method was developed and tested using the proposed method for location NEs that were inferred from 100 tweets written in colloquial Arabic. For evaluation purposes, the actual geocode coordinates were manually recorded to calculate the distance between actual and extracted geocode coordinates for each tweet. The results show that the proposed location NEL method locate 56.8% of tweets with a distance range of 0 – 10 km from the actual location. Further analysis has shown that the accuracy in locating tweets in an actual city and region are 78.9% and 84.2% respectively, and most tweets are located at distances greater than 100 km from the actual location referring to POI locations in different cities or regions, instead of the actual locations. On the other hand, the location NEL method returned NA geocode coordinates values for 5.3% of tweets. In the next chapter, a real-time flood detection system will be presented to help crisis management officers to identify and track floods in real-time through the exploitation of twitter and rainfall data. The next chapter explains how to develop a flood detection system by compiling the proposed NET and NEL methods.

# Chapter 8: A system for Real-Time Flood Detection

## 8.1 Introduction

In this chapter, the flood events detection approach will be extended to include live rainfall data. In particular, the proposed system aims to combine tweet data extracted from Twitter and rainfall data collected from the National Oceanic and Atmospheric Administration (NOAA). The proposed system has two goals: first, detect and locate floods mentioned by Twitter users; second, collect live rainfall data and study the possibility of predicting flooding events by analysing this Twitter and NOAA data. A second case study in a practical area (Makkah region, Saudi Arabia) was investigated to discuss the possibility of detecting a flood event before it happens from historical rainfall data and Twitter data. Makkah region was selected depending on recent research (Youssef et al., 2016) which has reported that the western and south-western regions of Saudi Arabia received the largest rainfall amount, which is where Makkah region is located.

Many efforts have been devoted to predicting floods in their early stages by utilising rainfall data. (Idate and Deshmukh, 2017) developed a real time floods forecasting system using the station's historical data. (Yang et al., 2015) applied artificial intelligence (AI) techniques to identify the rainfall threshold using different data sources, including historical flooding observations and rainfall data, to provide early warnings for flash floods caused by typhoons. A recent state-of-the-art review (Amezquita-Sanchez et al., 2017) covered different floods prediction works, and shows that there is a need to utilise different types and sources of data by applying big data technologies to develop flood prediction models. Furthermore,

they state that rainfall data which is collected using satellite technology and monitoring stations have proven to be useful sources of data to predict floods.

Predicting floods in the early stages will not be successful without proper data and information. As discussed in previous paragraphs, most systems or methods utilise rainfall data and other data sources; however, none of those studies have applied social media data to predict floods. This identified gap motivated us to formulate the following assumption:

***Assumption:*** it is possible to predict a flooding event in its early stages for a particular location by utilising different types of data sources, including historical rainfall data and Twitter data.

As reported by several authors (Youssef et al., 2016, Mashael Al, 2010, Deng et al., 2015), the most high risk floods in urban areas in Saudi Arabia are as a result of receiving rainfall runoff from the hills and mountains through different drainage and natural pathways valleys during heavy rainfall events. By building a data set which is based on rainfall data and Twitter data, it could be possible to track floods' pathways and detect flood sources.

## 8.2 Systems in Real-Time Flood Detection

In order to illustrate the proposed flood detecting system, a conceptual diagram is shown in Figure 4-2 (section 4.3). The architecture of the detection system consists of five components: tweets loader and pre-processing, classification, event location NER, location NEL and event visualisation, as shown in Figure 4-2. The following subsections provide the details of each component.

### 8.2.1 Microblog Loader and Pre-processing

A tweets loader has been developed to collect Twitter messages from public users via the Twitter API service. The user's initial query (i.e., a set of floods keywords) has been used

to collect all available tweets related to floods events. Twitter API services respond to user queries by sending related tweets and its metadata. The pre-processing stage includes tweet filtering based on tweet metadata, which includes the tweet's published time, tweet language, tweet source, whether it is a retweet, if it is a reply, tweet, geotagging information and attached images. On the other hand, second tweet filtering is based on tweet content, which includes removing punctuation marks and numbers, removing all non-Arabic characters, removing diacritics, letter normalisation, excluding a stop words list, and applying light stemming.

### 8.2.2 Tweet location NER and NEL

In this stage, the L2S method was applied to infer flood location NE from tweets that are classified as tweets mentioning high risk flood events, by a trained SVM classifier. After that, the proposed system utilised the location NEL method, which is explained in section 7-3, to extract the tweet geocode coordinates (lat, long) for each tweet. All inferring data and tweet metadata will be stored on a database to use in the future to predict floods in the early stages. The results of this stage will include only tweets with geocoding information and metadata.

### 8.2.3 Floods event visualisation

In this stage, an interactive web app was built straight from R language by utilising Shiny package. Shiny package allows us to build automatic "reactive" binding between inputs and outputs, and extensive prebuilt widgets make it possible to build beautiful, responsive, and powerful applications with minimal effort.

The interactive app includes three components, as shown in Figure 8-2, which are an interactive map, input slot and tweet metadata table. A "leaflet" package (Joe Cheng, 2017)

was installed, which provides an interactive map and includes interactive panning and zooming, allowing for an explorative view of the pinpointed location. Leaflet functions provide a simple and fast way to host interactive maps online in R and, therefore, have great integration in a workflow with Shiny.

Furthermore, this stage included collecting data on rainfall amounts from the National Oceanic and Atmospheric Administration (NOAA). NOAA is a scientific agency that provides several datasets and API related to weather data; “rnoaa” package (Scott et al., 2017) were installed, which provide daily precipitation data with 50 km resolution from the NOAA Climate Prediction Center (CPC).


| text | EventClass | long | lat | created |
|---|---|---|---|---|
| بالاستغفار ننعم بالامطار يارب أغثنا وأغث قلوبنا أول مرة ماحد يسأل جاتكم أمطار | positive | 45.08 | 23.89 | 2018-02-06 00:07:59 |
| أمطار خير وبركة اللهم أرضها في جميع بلاد المسلمين يارب | positive | 39.18 | 21.48 | 2018-02-06 02:03:12 |
| بالاستغفار ننعم بالامطار يارب امطار كثيره يارب | positive | 45.08 | 23.89 | 2018-02-06 02:48:37 |
| يارب امطار خير وبركه بالاستغفار ننعم بالامطار | positive | 45.08 | 23.89 | 2018-02-06 02:55:37 |
| سلام على الذين يزهرون القلوب اذا نزلوا بها و كأنهم بقاع القلب امطار | positive | 86.27 | 27.58 | 2018-02-06 03:12:31 |
| سلام على الذين يزهرون القلوب اذا نزلوا بها و كأنهم بقاع القلب امطار صباح الخير | positive | 86.27 | 27.58 | 2018-02-06 04:08:10 |
| الشتاء بينتهي ولاجانا امطار يارب لاتؤاخذنا بذنوبنا بالاستغفار ننعم بالامطار | positive | 45.08 | 23.89 | 2018-02-06 05:01:48 |
| سلام على الذين يزهرون القلوب اذا نزلوا بها وكأنهم في بقاع القلب امطار | positive | 44.83 | 13.76 | 2018-02-06 05:34:06 |
| سلام على الذين يزهرون القلوب اذا نزلوا بها وكأنهم في بقاع القلب أمطار صبأح الخير | positive | 48.52 | 15.55 | 2018-02-06 06:00:10 |
| بالاستغفار ننعم بالامطار الله يرزقنا أمطار ماتنقطع يارب | positive | 45.08 | 23.89 | 2018-02-06 06:39:34 |
| سلام على اللذين يزهرون القلوب اذا نزلوا بها وكأنهم في بقاع القلب امطار عمار العزكى | positive | 120.23 | 29.25 | 2018-02-06 |

*Figure 8-1 result screenshot*

## 8.3 Demonstration Scenario

For the demonstration, a collection of messages posted by Arabic Twitter users (specifically texts written in Arabic) were collected via the Twitter API service from 05 May to 01 Jun 2017 with the user's initial event-query used. Floods events were defined by specifying the keyword query (i.e., "سيول", "فيضانات", "سيل" and "أمطار قوية"). This time period was chosen because it was a time of heavy rain in Saudi Arabia, and people had started reporting and discussing flooding events. Also, the keywords which were most used by Twitter users to report a flood or heavy rain event were selected. The system was developed using R and Python languages and its valuable libraries. For a user, the system allows them to a enter time scale (i.e., one day, 10 hours or last 2 hours), which return tweets mentioning flood events during the specified time scale. After selecting a timescale and submitting the query, the system will start connecting with the Twitter API and then apply system processes, which are explained in section 8.2. The system processes take between one and two minutes to return a geotagged tweet and metadata table. The process time depends on the number of tweets returned by the Twitter API. As shown in Figure 8-3, interactive map bubbles are displayed over geographical regions with the tweet number located in that region. OSM provides map tiles at various zoom levels, starting from 0 to 18, so a user can track the accurate tweet location by using the zoom features, which will split the bubbles into smaller bubbles or into tagged tweets.

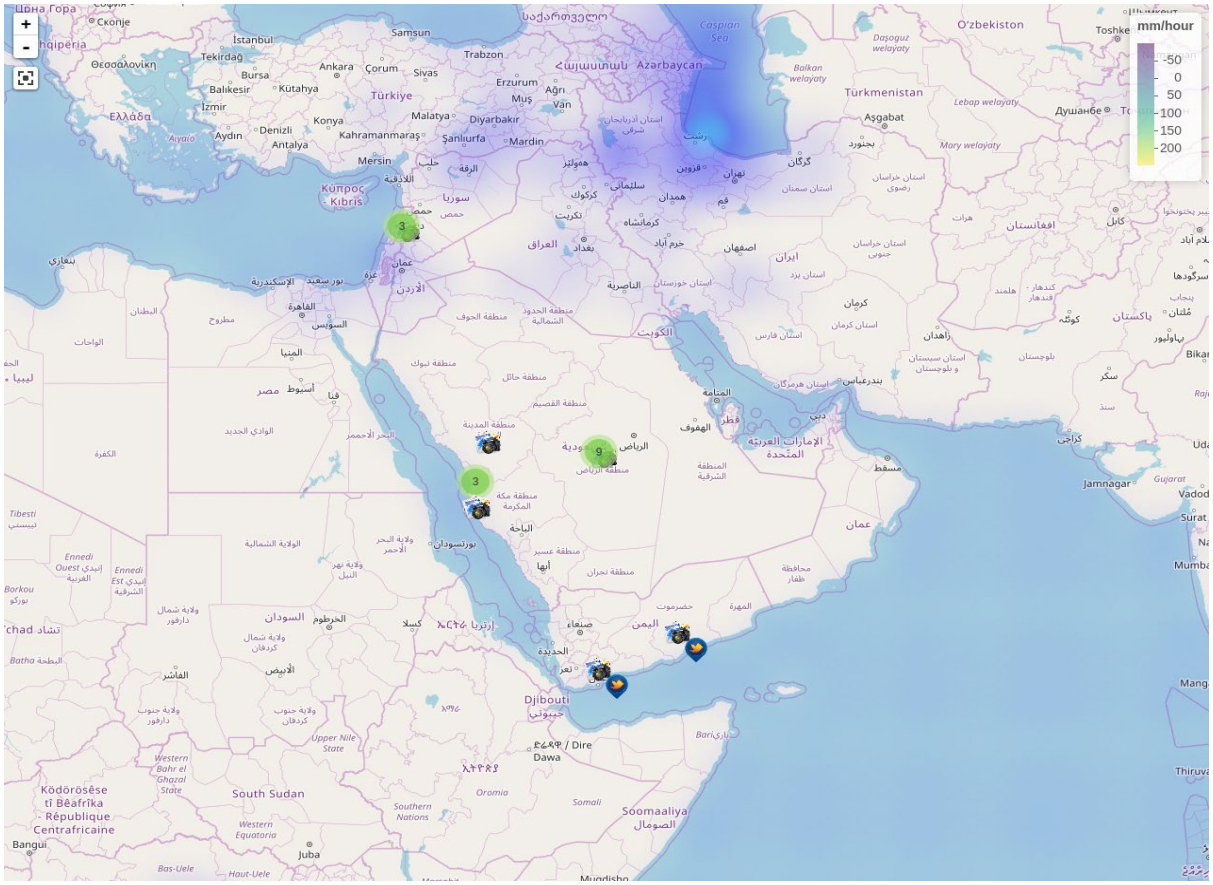


*Figure 8-2* *Interactive map screenshot*

Furthermore, it is possible to show tweet information and attach pictures by clicking the tweet marker. The maps will show the daily precipitation amount that has accumulated within the last 24 hours. The system also returns a table that lists all of the collected tweets which mention high risk flood events and their metadata. The system, by default, collects tweets published in the last five minutes every five minutes. The maps and table provided will help users or decision makers to view flood event information and act in real time to help reduce the significant damage that may occur as a result of that flood event.

## 8.4 Run Time Performance

As well as system accuracy, overall performance, it is also essential to consider metrics. Therefore, this section will measure the latency throughput of the proposed system, as it has

been found that this system can process 5000 tweets during a median time span of 37.32 seconds.

Latency has been measured according to the time delta from when the tweets were received until the proposed system returned an interactive map along with the related meta data. Table 8-1 presents the breakdown in the processing time for a sample of 5,000 tweets. Aggregation of the tweets was found to take 2.85 seconds from parsing the tweets from Twitter's API format and to returning a data frame containing the related meta data. It took 4.23 seconds for the pre-processing stage and to return a data frame containing the filtered tweets. During the classification stage, each tweet is either classified as positive (in relation to high flood levels) or negative (as it does not mention flooding), and this takes 6.67 seconds. The NER stage takes 9.59 seconds in identifying the event location for each tweet. The NEL stage requires the largest overheads in the system, with a median run time of 12.17 seconds. Most of the processing time during this stage is as a result of network delays while waiting for google API to respond. The 1.81 seconds remaining are required to present the interactive map and tweets meta data table.

*Table 8-1 proposed system processing time for a sample of 5,000 tweets*

| Component | Median Run Time (seconds) |
|---|---|
| Tweets aggregation | 2.85 |
| Pre-processing stage | 4.23 |
| Classification stage | 6.67 |
| NER stage | 9.59 |
| NEL stage | 12.17 |
| Visualization stage | 1.81 |
| Total | 37.32 |

## 8.5 Verifying and extending the system

This section aims to show the proposed system's results and reliability by examining the correlation between rainfall data and Twitter messages in a specific location. This correlation

has been evaluated by conducting a case study in the city of Makah, Saudi Arabia, using a dataset of rainfall data provided by NOAA. During the period from 05 May to 01 Jun 2017, around 10,454 tweets mentioning high risk floods were collected. The proposed system inferred the location for 6018 tweets; 2082 tweets were located within the Makah region (geocode coordinates located within rectangle bounded by (22.00,39.60) from the northwest and (21.04,41.59) from the southeast.

Figure 8-3 and Figure 8-4 show the daily rainfall amount for Arabian Peninsula, and the collected tweets were classified as positive (related to high floods) or negative (does not mention floods). Figure 8-3 and Figure 8-4 show that positive relationships were observed between rainfall amount and the number of tweets in the positive class. Moreover, it is clear that the number of tweets classified as negative increased, and the number of positive tweets decreased when the rainfall amount decreased, as recorded from 25 May until 01 Jun 2017. This indicator may be considered as evidence of the classifier's performance, taking into account that desert is the most prominent feature of the Arabian Peninsula, and more than half the area of Saudi Arabia is desert; as a result of that, most floods occur in desert and might not be observed by Twitter users. For that reason, an urban area (Makkah region, Saudi Arabia) has been adopted to obtain clear results.

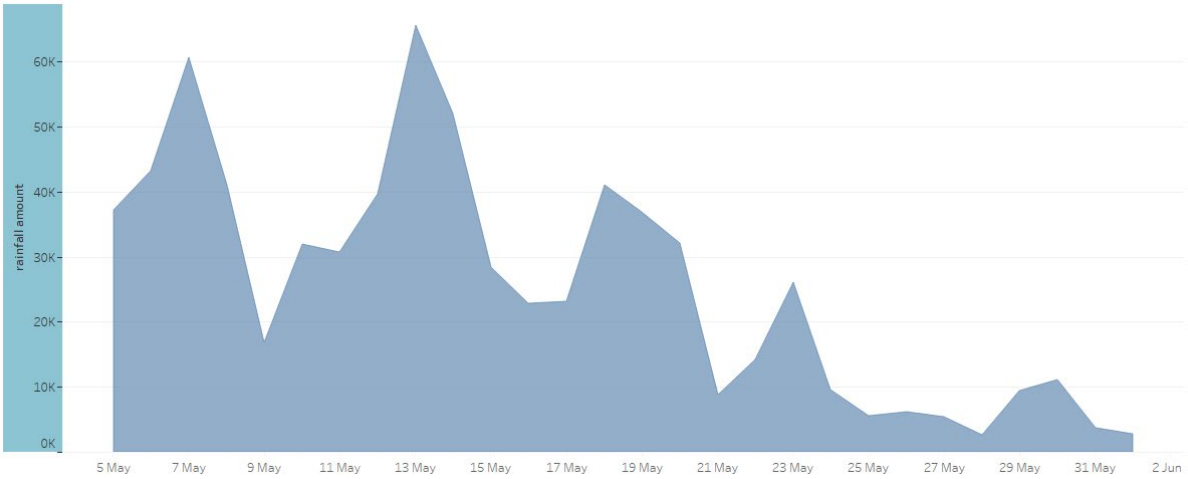


*Figure 8-3 Rainfall daily amount for Arabian Peninsula during time period between 5 May and 1 Jun 2017*

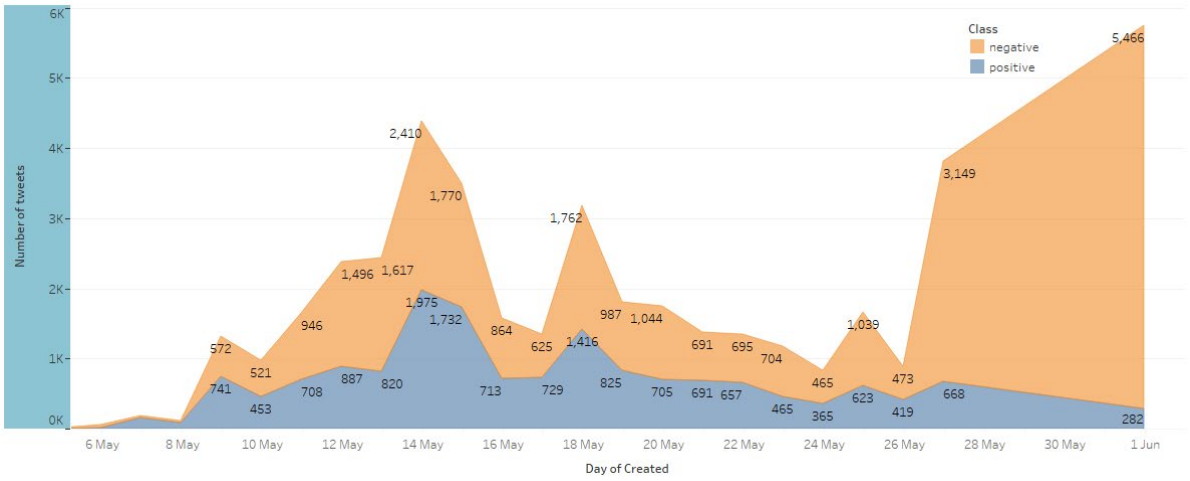


*Figure 8-4 number of tweets by class during time period between 5 May and 1 Jun 2017*

Figure 8-5 depicts, to a limited extent, the increase in rainfall and the number of tweets that mentioned high risk floods during the period 05 May - 01 Jun 2017, in Makah, Saudi Arabia, during the peak of the rainy season. As can be seen, there are similarities (based on the peaks) between the rainfall time-series and tweet time-series. However, there is not an exact correspondence between the time-series, which is due to the fact that floods occur after the rain, and some floods occur as a result of rain in uninhabited places such as mountains.

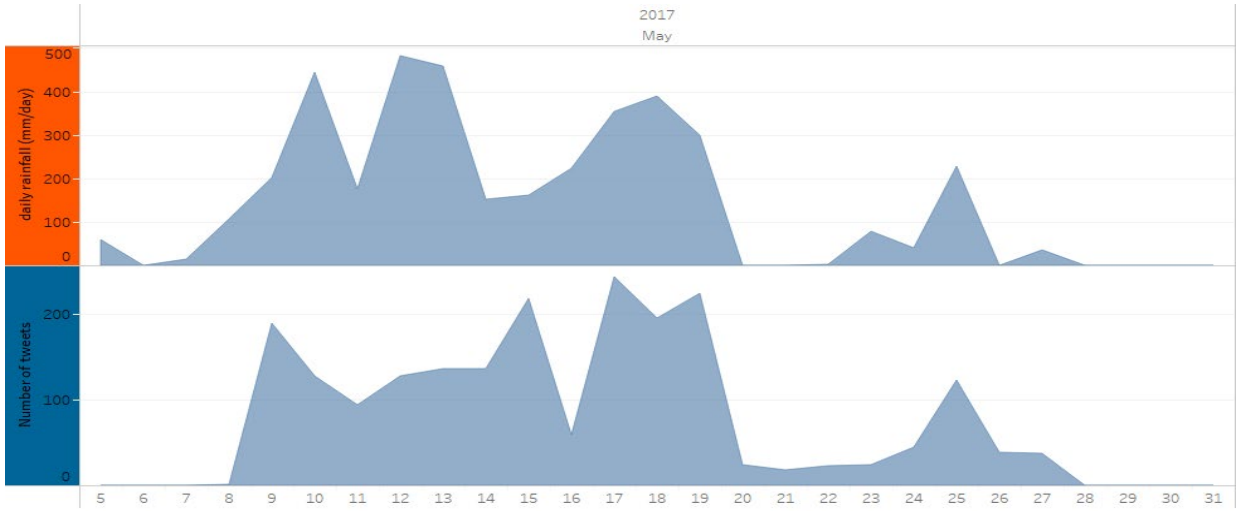


*Figure 8-5 daily rainfall amount received in Makkah region (above), and number of tweets are located in Makkah region (below) during time period between 5 May and 1 Jun 2017.*

It turns out that the indicators given by Figure 8-4 and Figure 8-5 reveal the possibly of utilising Twitter data and rainfall data to predict floods in the early stages, therefore developing a flood prediction system by using Twitter data should be considered in future work.

Several studies have discussed flood detection from twitter, however few of them have developed a flood detection system by using social media data. A recent study (de Bruijn et al., 2018) developed a global multi-lingual system (Tag, 2018) to detect natural disasters, including floods, in real time. Their system analyses tweets written in 12 languages which include English, Indonesian, Filipino, French, German, Italian, Polish, Serbian, Spanish and Turkish. The research carried out by (Holderness and Turpin, 2015) involved developing a real time flood detection system (petabencana, 2018)by analysing tweets written in the English or Indonesian language. However, neither of those floods detection systems are applicable to the Arabic language, even though it is the sixth most used language on twitter and highest number of AUs on Twitter are from Saudi Arabia (Aslam, 2017). It can be stated that with the huge number of tweets written in Arabic, the proposed flood detection system will add a valuable contribution as it deals with Arabic text. Furthermore, it utilises rainfall data, which is not used in other flood detection systems.

A Pearson correlation coefficient (Benesty, 2009) has been calculated in order to assess the relationship between the actual amount of daily rainfall and the number of tweets which mentioned high risk flooding between the 5th of May and the 1st of June 2017, in Makkah, Saudi Arabia. A positive correlation between the two variables was found: $r = 0.774$, $n = 28$, $p = 1.319^{e-06}$. In addition, significant evidence ($p < 0.001$) was found for an association between the number of tweets and the level of daily rainfall. Figure 8-6 presents a scatterplot graph that illustrates a summary of the results. A strong, positive correlation was found between

the amount of daily rainfall and the number of tweets. In addition, correlations were found between increases in daily rainfall amounts and increases in tweets mentioning high risk floods (see Appendix c for details of rainfall and tweets data).

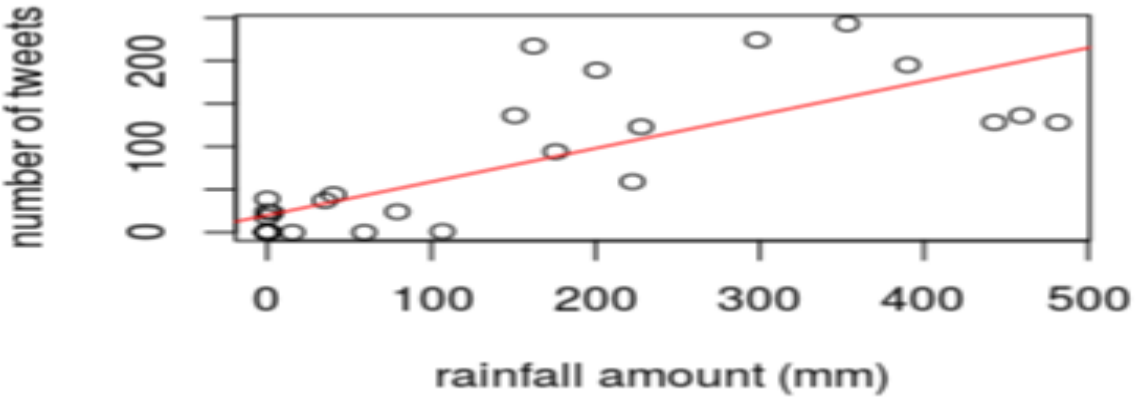


*Figure 8-6 scatterplot shows the daily rainfall amount and number of tweets data.*

## 8.6 Chapter summary

To summarise, this chapter has presented a real-time flood detection system and the results of its implementation. The architecture of the proposed detection system and its main components has been set out. The previous chapters have discussed and shown the need for flood detection by utilising Twitter users as sensors, especially Arabian users. In this chapter, as well as developing a flood detection system, a very good indicator of the possibility of utilising Twitter users as flood sensors to predict floods in the early stages has been shown. The proposed flood detection system has two unique features that do not exist in other floods detection systems, which are: (i) detect flooding events from Arabic tweets (ii) utilise rainfall data.

# Chapter 9: Conclusions and Future Research Directions

This chapter firstly summarises the main contributions to the research field and reviews the research findings with respect to the research problem statement. Furthermore, the practical consequences of this research are summarised. Finally, future directions for the research work described in this thesis are also discussed.

## 9.1 Summary of Contributions

The research started by setting the scene on the related background, problem statement, motivation and research scope, as stated in Chapter1. The thesis has defined the events and highlighted the event types and detection tasks from a social media perspective. This research on real-time flood events detection may be classified as an intersection between a specified event (event type) and new event detection (detection task). It has been clearly identified in sections 1.3 and 1.4 that there is a great need for a real-time flood detection system for crisis management in the Arabian Peninsula. The thesis has also highlighted, in section 1.2, the limitations of the existing work on flood detection using social media, and has listed the challenges that were faced in this current research. The thesis has also described the limited amount of research on specific event detection using social media platforms, especially from content written in colloquial Arabic.

This research contributes to the formalisation of the flood event detection problem from a scientific perspective, and it states the main research challenges and provides a standard evaluation framework that will allow information retrieval and NLP research communities to explore and compare solutions for the challenges mentioned.

Following this, Chapter 3 presented a critical review of the literature, which helped in designing the research methodology. The main purpose of the literature review was to identify key models, approaches and techniques that are used for event detection using social media and related areas to support decision making in crisis management. Therefore, the literature review was conducted on three different areas: text classification and its techniques in general; SLR to summarise the key characteristics of the different TC techniques and the methods used to classify Arabic text by utilising a strictly search protocol, and the third area covered approaches and techniques used for event detection and location inferring from social media platforms, specifically from twitter. Drawing on the findings from the preliminary and systematic review of the literature, a research design was proposed for a system for real-time flood detection in the Arabian Peninsula by utilising data mining and machine learning techniques.

This research has contributed to the formal definition of the main flood detection tasks: data collection from Twitter API to answer the question: Do tweets mention floods as keywords? A filtering task (what are the most influential filtering, or pre-processing, steps? A classification task (Does the tweet report real time floods events?) Extracting the location of floods from tweet content (How can flood locations be estimated accurately by using machine learning approaches?) A location named disambiguation task (How can extracted flood event locations be linked to an associate instance in a knowledge base?) And utilisation of rainfall data (How can trusted and real time rainfall data be capitalised on in flood detection systems?)

Besides the formalisation of the flood event detection problem, work specific to this research has focused on the filtering, classification and location NER and NEL tasks in analysing tweets written in colloquial Arabic text.

With respect to the filtering and classification tasks, this thesis has:

- Presented a comparative study that has discovered the most commonly used techniques of the pre-processing and classification stage to detect high-risk flood events in real-time.

- Classified task experiments utilising real world data which has been collected from the Twitter API to study the impact of stemming techniques on classifiers' performance, and to determine which classifier is most suitable for dealing with colloquial Arabic text. The experiments were implemented using different stemming techniques, which include Light 10 stemmer, words with the removal of common prefixes, and words with the removal of common suffixes in the Arabic language, and the most commonly used techniques for classifying MSA text, which included C5.0, J48, k-NN, NB, NNET and SVM. The results show that: (i) based on McNemar's statistical test, most classifiers perform better without applying stemming techniques when applied to colloquial Arabic text; (ii) The SVM classifier outperforms other classifiers in term of F1 measures.

- To the best of our knowledge, none of the previous studies have tried to produce a comparative study to study the impact of stemming trephines on colloquial Arabic text that covers all of these classifiers.

- An effective method for flood detection form Arabic tweets by using supervised learning techniques has been proposed.

The above contributions have addressed RQ1 by providing an effective method to extract tweets reporting high risk floods from Twitter data.

With respect to the task of location inferring from twitter, this research has:

- Developed an effective method for inferring and distinguishing flood location NE from colloquial Arabic tweets.

- Evaluated the proposed location NER method by comparing this method's results with existing near state-of-the-art Arabic NER systems. The results show that: (i) existing NER systems underperform when dealing with colloquial Arabic text compared with their performance when dealing with MSA; (ii) Existing NER systems (FARASA and Polyglot-NER) are not designed to distinguish event location from other locations mentions in a tweet; (iii) The proposed location NER method outperformed the FARASA and Polyglot-NER systems with significantly higher accuracy in tasks inferring flood locations from tweets which are written in colloquial Arabic.

The above contributions have addressed RQ2 by developing an effective method to infer event location from colloquial tweets.

With respect to the task of Locations Named Entity Linking, this research has:

- Proposed a competitive location NEL method that uses Google geocode API as a knowledge base to extract coordinates for inferred flood locations NE.

- Tested the proposed location NEL method on Arabic tweets, and the results show that: (i) the proposed system achieved a completeness degree of 94.7%; (ii) 56.8% of tweets in the test set are located within 10 km from the actual location; (iii) Taking into consideration that the maximum length of radius of cities and regions

in the Arabian Peninsula are 50 and 100 km respectively, the accuracy of locating a tweet in an actual city and region are 78.9% and 84.2% respectively; (iii) The proposed location NEL competes with other location NELs by taking into consideration that this method analyses colloquial Arabic text and focuses on the locations mentioned in the tweets' content.

RQ3 has been addressed by developing a location NEL method that located more than 78% of tweets in an actual city, as shown in the above contributions.

With respect to the Real-Time Floods Detection Systems, this research has:

- Proposed and implemented a real-time flood detection system by mining Twitter for crisis management in the Arabian Peninsula. The proposed system is language independent, satisfies the real-time requirement, and is suitable for a huge quantity of data.
- Compared the proposed flood detection system with existing systems developed by research institutes, and has found that the proposed system analyses tweets which are written in Arabic and utilises precipitation data; on the other hand, other systems are not applicable with Arabic text and have not used precipitation data.
- Presented a very good indicator that social media data might be valuable data for flood prediction when associated and integrated with different data sources such as precipitation data.

The proposed system, which is presented in Chapter 4, has addressed RQ4 by utilising rainfall data from NOAA and integrating it with Twitter data.

It is clear that a large number of events can be detected by utilising information from microblogs. Several studies have explored the use of trending topics and events to predict the impact of events, yet not many have addressed natural disasters, including flood events. Therefore, this thesis describes and analyses a flood detection system in order to address this gap and contribute to the knowledge in the area of text mining and machine learning by using microblog data that has been collected from Twitter.

A tweet is a type of document with unique characteristics; in particular, it is short, which makes it highly topic-focused. In addition, news-related tweets on a certain topic have a close correlation with temporal information. Twitter users often share event information, as well as personal views and perspectives when using Twitter to engage in online discussion. This research has led to a novel flood event detection system that is based on sequential pattern mining, and it has considered the characteristics of Twitter. The flood event detection system is made up of five components, which are: A Micro-blog Loader, Pre-processor, Classifier, Location Detector, and an Event Visualiser. Most of previous research on event detection and microblogs has been based on NER techniques, however, these do not distinguish the event location NE from other NE locations mentioned, making them more sensitive to noise and leading to a lack of clarity. Furthermore, location mismatches have occurred, which greatly affects retrieval and detection. For NE linking, an algorithm has been presented which uses Google API as the knowledge base, and this links the NE extracted to a specific spatial entity. Tweets that mention location NE can be very misleading, for example by referring to different types of spatial locations (e.g. points of interest, streets, cities, regions and countries); therefore, the algorithm in the current research has been designed to address this.

It is also important to note that previous flood detection systems have not used or analysed rainfall data, even though this is an important source for detecting and predicting flood events. Therefore, an interactive map has been presented which reveals the amount of rainfall that can improve the performance of flood detection systems and aid decision makers in monitoring flood events.

This research has a number of limitations that should be addressed in future studies, which are as follows:

- Real-time properties have not been systematically tested, in particular the access rate to the repository throughout the analysis tasks has not been noted, and the reachable crawling and processing rates have not been systematically collected. Therefore, the performance of this system has not been compared in detail with other similar systems.

- The system that has been proposed is based on language-independent models, making it possible to conduct the system by using tweets written in various languages; although a language expert would be needed to analyse the classifier learning data, as well as to annotate NE in every language used.

- In addition to the event types discussed previously in section 1.1.3, a number of detection tasks are required, such as NED; RED, and unspecified and specified event detection, bearing in mind that real time flood detection is classified as NED and specified event detection. The system that has been proposed is limited to detecting

specific events due to its design being based on machine learning techniques, therefore to detect alternative types of events (e.g earthquakes or hurricanes), it would be necessary to formulate and test a new data set for each event.

While there are several limitations, as mentioned above, this had a minor impact on the research, although these limitations should be addressed in future research as a way of building a robust and effective system.

## 9.2 Future Directions

This section lists the possible directions for future work that has the potential for further investigation into flood event detection using social media. It is hoped that the research presented in this thesis will lead to filling a gap identified in the research related to flood event detection and monitoring in social networks, and will encourage constructive future work in this area. There are various ways to extend the work in this thesis, which are listed below:

- Explore and investigate other pre-processing and classification techniques

One of the directions of this research is to explore and investigate additional pre-processing and classification techniques. This research has investigated how light stemming affects the classifier's performance when classifying colloquial Arabic text. In the future, we could study the effect of other pre-processing techniques such as n-gram (character level) or (word level) to improve classifier performance as much as possible. Regarding the classification step, additional classification models to apply to colloquial Arabic text could be explored.

- Explore and investigate other Named Entity Recognition tools

This research has applied the L2S method to address NER task; therefore, another direction is to consider deep learning NER algorithms such as Word2vec, word2vec and GloVe to improve the performance of the recognition of location NE in Arabic tweets. It could also be possible to implement part of speech tagging instead of a BIO tagging scheme.

- Explore and investigate another location knowledge base

Regarding future work on location NEL, attempting to improve the performance of the completeness and positional accuracy of the extracted coordinates by proposing a new location NEL approach that incorporates several location knowledge bases, such as Google, OSM and Wikipedia.

- Extend the proposed system to different languages or event types

As shown in section 8.2, the proposed system was designed by using supervised learning techniques including a classification task and NER task. However, the flood detection system does not utilise language-dependent models such as NER methods based on lexical or morphological features, therefore, with training data from different languages, it would be possible to implement the proposed system using tweets written in other languages. However, this would require a language expert to analyse the classifier learning data and to annotate NE for every language. Besides the event types, as discussed in section 1.1.3, there are several detection tasks, including NED, RED, unspecified and specified event detection; real time flood detection is classified as NED and specified event detection. Based on the fact that the proposed system can detect the event and infer the location of the event by analysing Twitter content, with proper training data, this system could be extended to detect other

specific spatial events that Twitter users mention, and event location, such as earthquakes, hurricanes and social events. The results presented in this thesis are related to detecting floods (as events) and Arabic (as the language), and applying other specific spatial events or languages would produce different results.

- Develop a flood prediction system for early stage crisis management.

As reported in a recent flood prediction review (Amezquita-Sanchez et al., 2017), there is a need to utilise different types and sources of data. To fill this identified gap, researchers could incorporate rainfall data which includes station and satellite data and social media data to develop flood prediction systems.

# References


ABABNEH, J., ALMOMANI, O., HADI, W., EL-OMARI, N. K. T. & AL-IBRAHIM, A. 2014. Vector Space Models to Classify Arabic Text. *International Journal of Computer Trends and Technology (IJCTT),* 7**,** 219-223.

ABAINIA, K., OUAMOUR, S. & SAYOUD, H. Topic identification of Arabic noisy texts based on KNN. Information and Communication Technology Research (ICTRC), 2015 International Conference on, 2015. IEEE, 92-95.

ABBAS, M., SMAILI, K. & BERKANI, D. 2010. Tr-classifier and knn evaluation for topic identification tasks. *The International Journal on Information and Communication Technologies (IJICT),* 3**,** 65-74.

ABDALLAH, S., SHAALAN, K. & SHOAIB, M. 2012. Integrating rule-based system with classification for arabic named entity recognition. *Computational Linguistics and Intelligent Text Processing***,** 311-322.

Abdelatti, Hillo, Yasin Elhadary, and Abbas Altayeb Babiker. "Nature and Trend of Urban Growth in Saudi Arabia: The Case of Al-Ahsa Province–Eastern Region." Resources and Environment 7.3 (2017): 69-80.

ABDULLA, N. A., AHMED, N. A., SHEHAB, M. A. & AL-AYYOUB, M. Arabic sentiment analysis: Lexicon-based and corpus-based. Applied Electrical Engineering and Computing Technologies (AEECT), 2013 IEEE Jordan Conference on, 2013. IEEE, 1-6.

AIELLO, L. M., PETKOS, G., MARTIN, C., CORNEY, D., PAPADOPOULOS, S., SKRABA, R., GÖKER, A., KOMPATSIARIS, I. & JAIMES, A. 2013. Sensing trending topics in Twitter. *IEEE Transactions on Multimedia,* 15**,** 1268-1282.

AJAO, O., HONG, J. & LIU, W. 2015. A survey of location inference techniques on Twitter. *Journal of Information Science,* 41**,** 855-864.

AL ZAGHOUL, F. & AL-DHAHERI, S. 2013. *Arabic Text Classification Based on Features Reduction Using Artificial Neural Networks*, Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on. IEEE, 2013.

AL-ANZI, F. S. & ABUZEINA, D. 2016. Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing. *Journal of King Saud University-Computer and Information Sciences*.

AL-BADARNEH, A., AL-SHAWAKFA, E., BANI-ISMAIL, B., AL-RABABAH, K. & SHATNAWI, S. 2016. The impact of indexing approaches on Arabic text classification. *Journal of Information Science***,** 0165551515625030.

AL-DIABAT, M. 2012. Arabic text categorization using classification rule mining. *Applied Mathematical Sciences,* 6**,** 4033-4046.

AL-HARBI, S., ALMUHAREB, A., AL-THUBAITY, A., KHORSHEED, M. S. & AL-RAJEH, A. 2008. Automatic Arabic text classification. *In, Proceedings of The 9th International Conference on the Statistical Analysis of Textual Data, Lyon-, France.*

AL-JALOUD, F., HEZAM, R. B. & AOUN-ALLAH, M. Classifying Arabic web pages toolkit. Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics, 2012. ACM, 69.

AL-KABI, M., AL-SHAWAKFA, E. & ALSMADI, I. 2013. The Effect of Stemming on Arabic Text Classification: An Empirical Study. *Information Retrieval Methods for Multidisciplinary Applications***,** 207.

AL-RFOU, R. 2017. *polyglot* [Online]. Available: https://pypi.python.org/pypi/polyglot [Accessed 15/09/2017].

AL-RFOU, R., KULKARNI, V., PEROZZI, B. & SKIENA, S. Polyglot-NER: Massive multilingual named entity recognition. Proceedings of the 2015 SIAM International Conference on Data Mining, 2015. SIAM, 586-594.

AL-SABBAGH, R. & GIRJU, R. YADAC: Yet another Dialectal Arabic Corpus. LREC, 2012. 2882-2889.

AL-SAGGAF, Y. 2012. Social media and political participation in Saudi Arabia: The case of the 2009 floods in Jeddah.

AL-SAUD, M. 2010. Assessment of flood hazard of Jeddah area 2009, Saudi Arabia. *Journal of Water Resource and Protection,* 2010.

AL-SHALABI, R., KANAAN, G. & GHARAIBEH, M. Arabic text categorization using kNN algorithm. Proceedings of The 4th International Multiconference on Computer Science and Information Technology, 2006. 5-7.

AL-SHALABI, R. & OBEIDAT, R. Improving KNN Arabic text classification with n-grams based document indexing. Proceedings of the Sixth International Conference on Informatics and Systems, Cairo, Egypt,

2008. Citeseer, 108-112.

AL-SHAMMARI, E. T. Improving Arabic document categorization: Introducing local stem. Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on, 2010. 385-390.

AL-SHARGABI, B., AL-ROMIMAH, W. & OLAYAH, F. A comparative study for Arabic text classification algorithms based on stop words elimination. Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications, 2011. ACM, 11.

AL-THUBAITY, A., ABANUMAY, N., AL-JERAYYED, S., ALRUKBAN, A. & MANNAA, Z. 2013. The Effect of Combining Different Feature Selection Methods on Arabic Text Classification. *2013 14th Acis International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/distributed Computing (Snpd 2013)***,** 211-216.

AL-THUBAITY, A., ALANAZI, A., HAZZAA, I. & AL-TUWAIJRI, H. Weirdness Coefficient as a Feature Selection Method for Arabic Special Domain Text Classification. Asian Language Processing (IALP), 2012 International Conference on, 2012. 69-72.

AL-THUBAITY, A., ALMUHAREB, A., AL-HARBI, S., AL-RAJEH, A. & KHORSHEED, M. 2008. *KACST Arabic Text Classification Project: Overview and preliminary results*.

AL-THWAIB, E. 2014a. Support Vector Machine versus k-Nearest Neighbor for Arabic Text Classification. *International Journal of Sciences,* 3**,** 1-5.

AL-THWAIB, E. 2014b. Text summarization as feature selection for arabic text classification. *World of Computer Science and Information Technology Journal (WCSIT),* 4**,** 101-104.

AL-WEHAIBI, R. N. & KHAN, M. B. Investigate the Context Usage of Arabic Proverbs in Twitter. Cloud Computing (ICCC), 2015 International Conference on, 2015. IEEE, 1-8.

ALAA, E. 2008. A Comparative Study on Arabic Text Classification. *researchgate.net*.

ALLAN, J., CARBONELL, J. G., DODDINGTON, G., YAMRON, J. & YANG, Y. 1998. Topic detection and tracking pilot study final report.

ALLWEIN, E. L., SCHAPIRE, R. E. & SINGER, Y. 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of machine learning research,* 1**,** 113-141.

ALSAEDI, N. & BURNAP, P. Arabic event detection in social media. International Conference on Intelligent Text Processing and Computational Linguistics, 2015. Springer, 384-401.

ALSAEDI, N., BURNAP, P. & RANA, O. Sensing Real-World Events Using Arabic Twitter Posts. Tenth International AAAI Conference on Web and Social Media, 2016.

ALSALEEM, S. 2011. Automated Arabic Text Categorization Using SVM and NB. *Int.Arab J.e-Technol.,* 2**,** 124-128.

ALSALEEM, S. M. 2013. Neural Networks for the Automation of Arabic Text Categorization. *2013 International Conference on Computer Applications Technology (Iccat)*.

ALTHENEYAN, A. S. & MENAI, M. E. B. 2014. Naïve Bayes classifiers for authorship attribution of Arabic texts. *Special Issue on Arabic NLP,* 26**,** 473-484.

ALWEDYAN, J., HADI, W. E. M., SALAM, M. A. & MANSOUR, H. Y. Categorize arabic data sets using multi-class classification based on association rule approach. Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications, 2011. ACM, 18.

AL-SHAWAKFA, E., AL-BADARNEH, A., SHATNAWI, S., AL-RABAB'AH, K. & BANI-ISMAIL, B. 2010. A comparison study of some Arabic root finding algorithms. *Journal of the American Society for Information Science and Technology,* 61**,** 1015-1024.

Amaral, José Nelson. "About computing science research methodology." (2011).

AMEZQUITA-SANCHEZ, J., VALTIERRA-RODRIGUEZ, M. & ADELI, H. 2017. Current efforts for prediction and assessment of natural disasters: Earthquakes, tsunamis, volcanic eruptions, hurricanes, tornados, and floods. *Scientia Iranica,* 24**,** 2645-2664.

API *Twitter*, T. 2018. *Twitter API* [Online]. Available: https://developer.twitter.com/en/docs/tweets/search/overview/standard [Accessed 4/4/2018 2018].

ASLAM, S. 2017. *Twitter by the Numbers: Stats, Demographics & Fun Facts.* [Online]. Available: https://www.omnicoreagency.com/twitter-statistics/ [Accessed 21/09/2017].

ATEFEH, F. & KHREICH, W. 2015. A survey of techniques for event detection in twitter. *Computational Intelligence,* 31**,** 132-164.

AYEDH, A., TAN, G., ALWESABI, K. & RAJEH, H. 2016. The Effect of Preprocessing on Arabic Document Categorization. *Algorithms,* 9**,** 27.

BACKSTROM, L., SUN, E. & MARLOW, C. Find me if you can: improving geographical prediction with social and spatial proximity. Proceedings of the 19th international conference on World wide web, 2010. ACM, 61-70.

BASNUR, P. W. & SENSUSE, D. I. 2010. Pengklasifikasian Otomatis Berbasis Ontologi Untuk Artikel Berita Berbahasa Indonesia. *MAKARA of Technology Series,* 14.

BECKER, H., NAAMAN, M. & GRAVANO, L. 2011. Beyond Trending Topics: Real-World Event Identification on Twitter. *ICWSM,* 11**,** 438-441.

BELKEBIR, R. & GUESSOUM, A. 2013. A Hybrid BSO-Chi2-SVM Approach to Arabic Text Categorization. *2013 Acs International Conference on Computer Systems and Applications (Aiccsa)*.

BELLMORE, A., CALVIN, A. J., XU, J.-M. & ZHU, X. 2015. The five w's of "bullying" on twitter: who, what, why, where, and when. *Computers in human behavior,* 44**,** 305-314.

BEN OTHMANE ZRIBI, C., BEN FRAJ, F. & BEN AHMED, M. Combining classifiers for supertagging Arabic texts. Natural Language Processing and Knowledge Engineering (NLP-KE), 2010 International Conference on, 2010. IEEE, 1-9.

BENAJIBA, Y., DIAB, M. & ROSSO, P. Arabic named entity recognition using optimized feature sets. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2008. Association for Computational Linguistics, 284-293.

Benesty, Jacob, et al. "Pearson correlation coefficient." Noise reduction in speech processing. Springer, Berlin, Heidelberg, 2009. 1-4.

BOSTANCI, B. & BOSTANCI, E. An evaluation of classification algorithms using Mc Nemar's test. Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012), 2013. Springer, 15-26.

Bowden, J. (2014). Social Media APIs and Data Collection Strategies. [online] Available at: https://www.business2community.com/social-media/social-media-apis-data-collection-strategies-0887426 [Accessed 31 Sep. 2018].

BRAHIMI, B., TOUAHRIA, M. & TARI, A. 2016. Data and Text Mining Techniques for Classifying Arabic Tweet Polarity. *Journal of Digital Information Management,* 14**,** 15.

BROWN, P. F., DESOUZA, P. V., MERCER, R. L., PIETRA, V. J. D. & LAI, J. C. 1992. Class-based n-gram models of natural language. *Computational linguistics,* 18**,** 467-479.

CAN, F., KOCBERBER, S., BALCIK, E., KAYNAK, C., OCALAN, H. C. & VURSAVAS, O. M. 2008. Information retrieval on Turkish texts. *Journal of the American Society for Information Science and Technology,* 59**,** 407-421.

CHAKRAVARTY, S. A survey on text classification techniques for e-mail filtering. Machine Learning and Computing (ICMLC), 2010 Second International Conference on, 2010. IEEE, 32-36.

CHENG, Z., CAVERLEE, J. & LEE, K. You are where you tweet: a content-based approach to geo-locating twitter users. Proceedings of the 19th ACM international conference on Information and knowledge management, 2010. ACM, 759-768.

CLARK, A. F. & CLARK, C. 1999. Performance characterization in computer vision a tutorial*.) N (Eds.): FBook performance characterization in computer vision a tutorial/(Citeseer, 1999, edn.)*.

COHEN, W. W. & HIRSH, H. Joins that Generalize: Text Classification Using WHIRL. KDD, 1998. 169-173.

COHEN, W. W. & SINGER, Y. 1999. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems (TOIS),* 17**,** 141-173.

COTTERELL, R. & CALLISON-BURCH, C. A Multi-Dialect, Multi-Genre Corpus of Informal Written Arabic. LREC, 2014. 241-245.

CUNNINGHAM, H., MAYNARD, D., BONTCHEVA, K. & TABLAN, V. A framework and graphical development environment for robust NLP tools and applications. ACL, 2002. 168-175.

DAGAN, I., KAROV, Y. & ROTH, D. 1997. Mistake-driven learning in text categorization. *arXiv preprint cmp-lg/9706006*.

DARWISH, K. & GAO, W. Simple Effective Microblog Named Entity Recognition: Arabic as an Example. LREC, 2014. 2513-2517.

DAUMÉ III, H., LANGFORD, J. & ROSS, S. 2014. Efficient programmable learning to search. *arXiv preprint arXiv:1406.1837*.

DAVIS JR, C. A., PAPPA, G. L., DE OLIVEIRA, D. R. R. & DE L ARCANJO, F. 2011. Inferring the location of twitter messages based on user relationships. *Transactions in GIS,* 15**,** 735-751.

DBPEDIA. 2017. Available: http://wiki.dbpedia.org/ [Accessed 22/03/2018].

DE BRUIJN, J. A., DE MOEL, H., JONGMAN, B., WAGEMAKER, J. & AERTS, J. C. 2018. TAGGS: Grouping Tweets to Improve Global Geoparsing for Disaster Response. *Journal of Geovisualization and Spatial Analysis,* 2**,** 2.

DENG, L., MCCABE, M. F., STENCHIKOV, G., EVANS, J. P. & KUCERA, P. A. 2015. Simulation of flash-flood-producing storm events in Saudi Arabia using the weather research and forecasting model. *Journal of Hydrometeorology,* 16**,** 615-630.

DERCZYNSKI, L., MAYNARD, D., RIZZO, G., VAN ERP, M., GORRELL, G., TRONCY, R., PETRAK, J. & BONTCHEVA, K. 2015. Analysis of named entity recognition and linking for tweets. *Information Processing & Management,* 51**,** 32-49.

DICTIONARY. 2018. *event* [Online]. Available: http://www.dictionary.com/browse/event?s=t [Accessed 22/03/2018].

DIETTERICH, T. G. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation,* 10**,** 1895-1923.

DONG, X., MAVROEIDIS, D., CALABRESE, F. & FROSSARD, P. 2015. Multiscale event detection in social media. *Data Mining and Knowledge Discovery,* 29**,** 1374-1405.

DOVGOPOL, R. & NOHELTY, M. 2015. Twitter hash tag recommendation. *arXiv preprint arXiv:1502.00094*.

DUMAIS, S., PLATT, J., HECKERMAN, D. & SAHAMI, M. Inductive learning algorithms and representations for text categorization. Proceedings of the seventh international conference on Information and knowledge management, 1998. ACM, 148-155.

DUWAIRI, R., AL-REFAI, M. N. & KHASAWNEH, N. 2009. Feature Reduction Techniques for Arabic Text Categorization. *Journal of the American Society for Information Science and Technology,* 60**,** 2347-2352.

DUWAIRI, R. & EL-ORFALI, M. 2014. A study of the effects of preprocessing strategies on sentiment analysis for Arabic text. *Journal of Information Science,* 40**,** 501-513.

DUWAIRI, R. M. 2006. Machine learning for Arabic text categorization. *Journal of the American Society for Information Science and Technology,* 57**,** 1005-1010.

DUWAIRI, R. M. 2007. Arabic Text Categorization. *The International Arab Journal of Information technology,* 4**,** 125-132.

DUWAIRI, R. M. Statistical Feature Selection Techniques for Arabic Text Categorization. The Fourth International Conference on Information and Communication Systems (ICICS 2013), , 2013 Irbid, Jordan, April, 23-25,.

DYNES, R. R. 1970. *Organized behavior in disaster*, Heath LexingtonBooks.

EIBE FRANK, M. A. H., AND IAN H. WITTEN (2016) 2016. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques". Morgan Kaufmann.

EILANDER, D., TRAMBAUER, P., WAGEMAKER, J. & VAN LOENEN, A. 2016. Harvesting social media for generation of near real-time flood maps. *Procedia Engineering,* 154**,** 176-183.

EL-HALEES, A. M. 2007. Arabic text classification using maximum entropy. *The Islamic University Journal (Series of Natural Studies andEngineering)* Vol. 15,No.1,pp 157-167, 2007, ISSN 1726-6807, http//www.iugzaza.edu.ps/ara/research/.

ELBERRICHI, Z. & ABIDI, K. 2012. Arabic Text Categorization: a Comparative Study of Different Representation Modes. *International Arab Journal of Information Technology,* 9**,** 465-470.

ELLISON, N. B. 2007. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication,* 13**,** 210-230.

FEINERER, I., HORNIK, K. & FEINERER, M. I. 2015. Package 'tm'. *Corpus,* 10.

FINKEL, J. R., GRENAGER, T. & MANNING, C. Incorporating non-local information into information extraction systems by gibbs sampling. Proceedings of the 43rd annual

meeting on association for computational linguistics, 2005. Association for Computational Linguistics, 363-370.
FROMMHOLZ, I., AL-KHATEEB, H. M., POTTHAST, M., GHASEM, Z., SHUKLA, M. & SHORT, E. 2016. On Textual Analysis and Machine Learning for Cyberstalking Detection. *Datenbank-Spektrum,* 16**,** 127-135.
FUHR, N., HARTMANN, S., LUSTIG, G., SCHWANTNER, M., TZERAS, K. & KNORZ, G. AIR/X: A rule-based multistage indexing system for large subject fields. Intelligent Text and Image Handling-Volume 2, 1991. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 606-623.
GEIß, J., SPITZ, A. & GERTZ, M. NECKAr: A Named Entity Classifier for Wikidata. International Conference of the German Society for Computational Linguistics and Language Technology, 2017. Springer, 115-129.
GENTRY, J., GENTRY, M. J., RSQLITE, S. & ARTISTIC, R. L. 2016. Package 'twitteR'.
GOOGLE 2018a. Google Maps Geocoding API.
GOOGLE 2018b. Google Maps Geocoding API Usage Limits.
HABASH, N. Y. 2010. Introduction to Arabic natural language processing. *Synthesis Lectures on Human Language Technologies,* 3**,** 1-187.
HADI, W. E. 2015. Classification of Arabic Social Media Data. *Advances in Computational Sciences and Technology,* 8**,** 29-34.
HADI, W. E. M., SALAM, M. A. & AL-WIDIAN, J. A. Performance of nb and svm classifiers in islamic arabic data. Proceedings of the 1st International Conference on Intelligent Semantic Web-Services and Applications, 2010. ACM, 14.
HADNI, M., LACHKAR, A. & ALAOUI OUATIK, S. 2012. A New and Efficient Stemming Technique for Arabic Text Categorization. *2012 International Conference on Multimedia Computing and Systems (Icmcs)***,** 791-796.
HAN, J., KAMBER, M. & PEI, J. 2011. *Data mining: concepts and techniques*, Elsevier.
HARRAG, F. & AL-QAWASMAH, E. 2010. Improving Arabic Text Categorization Using Neural Network with SVD. *JDIM,* 8**,** 233-239.
HARRAG, F. & EL-QAWASMAH, E. 2009. *Neural Network for Arabic Text Classification*, In Applications of Digital Information and Web Technologies, 2009. ICADIWT'09. Second International Conference on the (pp. 778-783). IEEE.
HARRAG, F., EL-QAWASMAH, E. & AL-SALMAN, A. M. S. Comparing Dimension Reduction Techniques for Arabic Text Classification Using BPNN Algorithm. Integrated Intelligent Computing (ICIIC), 2010 First International Conference on, 2010. 6-11.
He, He, Hal Daumé III, and Jason Eisner. "Dynamic feature selection for dependency parsing." Proceedings of the 2013 conference on empirical methods in natural language processing. 2013.
HIDAYATULLAH, A. F., RATNASARI, C. I. & WISNUGROHO, S. 2016. Analysis of Stemming Influence on Indonesian Tweet Classification. *TELKOMNIKA (Telecommunication Computing Electronics and Control),* 14**,** 665-673.
HIGGINS, J. P. T. & GREEN, S. 2008. *Cochrane handbook for systematic reviews of interventions*, Wiley Online Library.
HMEIDI, I., AL-AYYOUB, M., ABDULLA, N. A., ALMODAWAR, A. A., ABOORAIG, R. & MAHYOUB, N. A. 2014. Automatic Arabic text categorization: A comprehensive comparative study. *Journal of Information Science***,** 0165551514558172.

HMEIDI, I., HAWASHIN, B. & EL-QAWASMEH, E. 2008. Performance of KNN and SVM classifiers on full word Arabic articles. *Intelligent computing in engineering and architecture,* 22**,** 106-111.

HOLDERNESS, T. & TURPIN, E. 2015. White paper—PetaJakarta. org: Assessing the role of social media for civic co-management during monsoon flooding in Jakarta, Indonesia. *University of Wollongong, Wollongong*.

HOSPEDALES, T. M., GONG, S. & XIANG, T. 2013. Finding rare classes: Active learning with generative and discriminative models. *Knowledge and Data Engineering, IEEE Transactions on,* 25**,** 374-386.

Horvitz, Daniel G., and Donovan J. Thompson. "A generalization of sampling without replacement from a finite universe." Journal of the American statistical Association 47.260 (1952): 663-685.

HUANG, F. Improved Arabic dialect classification with social media data. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015. 2118-2126.

IBM. 2018. *alchemy-api* [Online]. Available: https://www.ibm.com/watson/alchemy-api.html [Accessed 22/03/2018].

IDATE, A. S. & DESHMUKH, R. 2017. Real Time Flood Forecasting System Using Artificial Neural Networks.

IKAWA, Y., VUKOVIC, M., ROGSTADIUS, J. & MURAKAMI, A. Location-based insights from the social web. Proceedings of the 22nd international conference on World Wide Web, 2013. ACM, 1013-1016.

INKPEN, D., LIU, J., FARZINDAR, A., KAZEMI, F. & GHAZI, D. Detecting and disambiguating locations mentioned in twitter messages. International Conference on Intelligent Text Processing and Computational Linguistics, 2015. Springer, 321-332.

INSTITUTE, Q. C. R. 2017. *FARASA* [Online]. Available: http://qatsdemo.cloudapp.net/farasa/ [Accessed 15/09/2017].

JEBARA, T. 2001. Discriminative, generative and imitative learning.

JOACHIMS, T. Text categorization with support vector machines: Learning with many relevant features. European conference on machine learning, 1998. Springer, 137-142.

JOE CHENG, B. K. A. Y. X. 2017. leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library. R package version

1.1.0. https://CRAN.R-project.org/package=leaflet.

JONGMAN, B., WAGEMAKER, J., ROMERO, B. R. & DE PEREZ, E. C. 2015. Early flood detection for rapid humanitarian response: harnessing near real-time satellite and Twitter signals. *ISPRS International Journal of Geo-Information,* 4**,** 2246-2266.

JORDAN, A. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems,* 14**,** 841.

KAATI, L., OMER, E., PRUCHA, N. & SHRESTHA, A. Detecting Multipliers of Jihadism on Twitter. 2015 IEEE International Conference on Data Mining Workshop (ICDMW), 2015. IEEE, 954-960.

KADHIM, M. H. & OMAR, N. 2012. Automatic Arabic Text Categorization using Bayesian Learning. *2012 7th International Conference on Computing and Convergence Technology (Iccct2012)***,** 415-419.

KANAAN, G., AL-SHALABI, R., GHWANMEH, S. & AL-MA'ADEED, H. 2009. A comparison of text-classification techniques applied to Arabic text. *Journal of the American Society for Information Science and Technology,* 60**,** 1836-1844.

KANAN, T. & FOX, E. A. 2016. Automated Arabic Text. Classification with P-Stemmer. *Machine Learning, and a Tailored News Article Taxonomy. J. Assoc. Inf. Sci. Technol*.

KHOJA, S. APT: Arabic part-of-speech tagger. Proceedings of the Student Workshop at NAACL, 2001. 20-25.

KHORSHEED, M. S. & AL-THUBAITY, A. O. 2013. Comparative evaluation of text classification techniques using a large diverse Arabic dataset. *Language Resources and Evaluation,* 47**,** 513-538.

KHREISAT, L. 2009. A machine learning approach for Arabic text classification using N-gram frequency statistics. *Journal of Informetrics,* 3**,** 72-77.

KITCHENHAM, B. & CHARTERS, S. 2007. Guidelines for performing systematic literature reviews in software engineering. *Technical report, Ver. 2.3 EBSE Technical Report. EBSE.*

KRISHNAMURTHY, S. & DOU, W. 2008. Note from special issue editors: advertising with user-generated content: a framework and research agenda. *Journal of Interactive Advertising,* 8**,** 1-4.

KRYVASHEYEU, Y., CHEN, H., OBRADOVICH, N., MORO, E., VAN HENTENRYCK, P., FOWLER, J. & CEBRIAN, M. 2016. Rapid assessment of disaster damage using social media activity. *Science advances,* 2**,** e1500779.

KUHN, M. 2008. Caret package. *Journal of Statistical Software,* 28**,** 1-26.

LAM, S. L. & LEE, D. L. Feature reduction for neural network based text categorization. Database Systems for Advanced Applications, 1999. Proceedings., 6th International Conference on, 1999. IEEE, 195-202.

LARKEY, L. S., BALLESTEROS, L. & CONNELL, M. E. 2007. Light stemming for Arabic information retrieval. *Arabic computational morphology.* Springer.

LATONERO, M. & SHKLOVSKI, I. 2011. Emergency management, Twitter, and social media evangelism.

LEMKE, D., MATTAUCH, V., HEIDINGER, O. & HENSE, H. 2015. Who hits the mark? A comparative study of the free geocoding services of Google and OpenStreetMap. *Gesundheitswesen (Bundesverband der Arzte des Offentlichen Gesundheitsdienstes (Germany)),* 77**,** e160-5.

LEWIS, D. D. & CATLETT, J. 1994. Heterogeneous uncertainty sampling for supervised learning. *Machine Learning Proceedings 1994.* Elsevier.

LEWIS, D. D. & RINGUETTE, M. A comparison of two learning algorithms for text categorization. Third annual symposium on document analysis and information retrieval, 1994. 81-93.

LI, C., SUN, A., WENG, J. & HE, Q. 2015. Tweet segmentation and its application to named entity recognition. *IEEE Transactions on Knowledge and Data Engineering,* 27**,** 558-570.

LI, R., WANG, S., DENG, H., WANG, R. & CHANG, K. C.-C. Towards social user profiling: unified and discriminative influence model for inferring home locations. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012. ACM, 1023-1031.

LI, Y. H. & JAIN, A. K. 1998. Classification of text documents. *The Computer Journal,* 41**,** 537-546.

LIU, X., LI, Y., WU, H., ZHOU, M., WEI, F. & LU, Y. Entity Linking for Tweets. ACL (1), 2013. 1304-1311.

Ma, Chao, et al. "Prune-and-score: Learning for greedy coreference resolution." Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014.

MAHAFDAH, R., OMAR, N. & AL-OMARI, O. 2014. Arabic Part of Speech Tagging Using K-Nearest Neighbour and Naive Bayes Combination *Journal of Computer Science,* 10**,** 1865-1873.

MAJED, I. H., FEKRY, O., MINWER, A. L. D. & SHAMSAN, A. 2011. Arabic Text Classification using Smo,naÏve Bayesian, J48 Algorithms. *International Journal of Research and Reviews in Applied Sciences*.

MAMOUN, R. & AHMED, M. A. 2014. A Comparative Study on Different Types of Approaches to the Arabic text classification. *1st International Conference of Recent Trends in Information and Communication Technologies*.

MASHAEL AL, S. 2010. Assessment of flood hazard of Jeddah area 2009, Saudi Arabia. *Journal of Water Resource and Protection,* 2010.

MCCALLUM, A. & NIGAM, K. A comparison of event models for naive bayes text classification. AAAI-98 workshop on learning for text categorization, 1998. Citeseer, 41-48.

MCNEMAR, Q. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika,* 12**,** 153-157.

MEDVET, E. & BARTOLI, A. Brand-related events detection, classification and summarization on twitter. Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on, 2012. IEEE, 297-302.

MESLEH, A. M. D. 2011. Feature sub-set selection metrics for Arabic text classification. *Pattern Recognition Letters,* 32**,** 1922-1929.

MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G. S. & DEAN, J. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 2013. 3111-3119.

MITCHELL, T. M. 1997. Machine learning. WCB. McGraw-Hill Boston, MA:.

MOHAMMAD, A. H., AL-MOMANI, O. & ALWADA'N, T. 2016. Arabic Text Categorization using k-nearest neighbour, Decision Trees (C4. 5) and Rocchio Classifier: A Comparative Study.

MONTEJO RÁEZ, A. 2005. Automatic text categorization of documents in the High Energy Physics domain.

MOSHFEGHI, M. 2016. Location-aware content and location-based advertising with a mobile device. Google Patents.

NEHAR, A., ZIADI, D. & CHERROUN, H. 2013. Rational kernels for arabic text classification. *Statistical Language and Speech Processing.* Springer.

NIELSEN, R. 2017. arabicStemR: Arabic Stemmer for Text Analysis.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine learning*, *39*(2-3), 103-134.

NOAMAN, A. & AL-GHURIBI, S. 2012. A NEW APPROACH FOR ARABIC TEXT CLASSIFICATION USING LIGHT STEMMER AND PROBABILITIES. *International Journal of Academic Research,* 4.

ODEH, A., ABU-ERRUB, A., SHAMBOUR, Q. & TURAB, N. 2015. Arabic Text Categorization Algorithm using Vector Evaluation Method. *arXiv preprint arXiv:1501.01318*.

OMER, M. A. H. & MA, S.-L. 2010. Stemming algorithm to classify Arabic documents. *Journal of Communication and Computer,* 7**,** 1-5.

ONTOTEXT. 2018. Available: https://ontotext.com/ [Accessed 22/03/2018].

ORANGE 2017. orange.
PAN, C.-C. & MITRA, P. Event detection with spatial latent Dirichlet allocation. Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries, 2011. ACM, 349-358.
PARALIC, J. & BEDNAR, P. 2003. Text mining for document annotation and ontology support. *Intelligent Systems at the Service of Mankind*, 237-248.
PAUL, M. J. & DREDZE, M. 2011. You are what you Tweet: Analyzing Twitter for public health. *Icwsm,* 20, 265-272.
PETABENCANA. 2018. Available: https://petabencana.id/map [Accessed].
POPESCU, A.-M. & PENNACCHIOTTI, M. Detecting controversial events from twitter. Proceedings of the 19th ACM international conference on Information and knowledge management, 2010. ACM, 1873-1876.
PORTE, J. A. A. J. H. A. V. M. A. N. 2017. markdown: 'Markdown' Rendering for R.
POULIQUEN, B., STEINBERGER, R. & IGNAT, C. 2006. Automatic annotation of multilingual text collections with a conceptual thesaurus. *arXiv preprint cs/0609059*.
RAHEEL, S. & DICHY, J. 2010. An Empirical Study on the Feature's Type Effect on the Automatic Classification of Arabic Documents. *Computational Linguistics and Intelligent Text Processing.* Springer.
RAINA, R., SHEN, Y., MCCALLUM, A. & NG, A. Y. Classification with hybrid generative/discriminative models. Advances in neural information processing systems, 2003.
RAPIDMINER 2018. rapidminer.
Rigutini, Leonardo, Marco Maggini, and Bing Liu. "An EM based training algorithm for cross-language text categorization." Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence. IEEE Computer Society, 2005
RITTER, A., CLARK, S. & ETZIONI, O. Named entity recognition in tweets: an experimental study. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011. Association for Computational Linguistics, 1524-1534.
Rao, Sudha, et al. "CLIP $@ $ UMD at SemEval-2016 Task 8: Parser for Abstract Meaning Representation using Learning to Search." Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). 2016.
RYOO, K. & MOON, S. Inferring twitter user locations with 10 km accuracy. Proceedings of the 23rd International Conference on World Wide Web, 2014. ACM, 643-648.
SAAD, M. K. & ASHOUR, W. Arabic text classification using decision trees. Proceedings of the 12th international workshop on computer science and information technologies CSIT, 2010. 75-79.
SAKAKI, T., OKAZAKI, M. & MATSUO, Y. Earthquake shakes Twitter users: real-time event detection by social sensors. Proceedings of the 19th international conference on World wide web, 2010. ACM, 851-860.
SALEM, F. 2017. Social Media and the Internet of Things towards Data-Driven Policymaking in the Arab World: Potential, Limits and Concerns.
SCHULZ, A., HADJAKOS, A., PAULHEIM, H., NACHTWEY, J. & MÜHLHÄUSER, M. A Multi-Indicator Approach for Geolocalization of Tweets. ICWSM, 2013.
SCHÜTZE, H., HULL, D. A. & PEDERSEN, J. O. A comparison of classifiers and document representations for the routing problem. Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, 1995. ACM, 229-237.

SCOTT, C., BROOKE, A., MAËLLE, S., ADAM, E., NICHOLAS, P., JOSEPH, S., ALEX, S., KARTHIK, R. & HART, E. 2017. 'NOAA' Weather Data from R.

SEBASTIANI, F. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR),* 34**,** 1-47.

Sharaf, Amr, and Hal Daumé III. "Structured prediction via learning to search under bandit feedback." Proceedings of the 2nd Workshop on Structured Prediction for Natural Language Processing. 2017.

SHOUKRY, A. & RAFEA, A. Sentence-level Arabic sentiment analysis. Collaboration Technologies and Systems (CTS), 2012 International Conference on, 2012. IEEE, 546-550.

SINGLA, A., PATRA, S. & BRUZZONE, L. 2014. A novel classification technique based on progressive transductive SVM learning. *Pattern Recognition Letters,* 42**,** 101-106.

SKORIC, M., POOR, N., ACHANANUPARP, P., LIM, E.-P. & JIANG, J. Tweets and votes: A study of the 2011 singapore general election. System Science (HICSS), 2012 45th Hawaii International Conference on, 2012. IEEE, 2583-2591.

SMITH, G. 2017. Number of social media users worldwide from 2010 to 2021 (in billions). Available: https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/.

SUN, A., LIM, E.-P. & LIU, Y. 2009. On strategies for imbalanced text classification using SVM: A comparative study. *Decision Support Systems,* 48**,** 191-201.

Tan, Songbo. "Neighbor-weighted k-nearest neighbor for unbalanced text corpus." *Expert Systems with Applications* 28.4 (2005): 667-671

TAG, F. 2018. Available: https://www.globalfloodmonitor.org/ [Accessed].

TEAM, R. C. 2013. R: A language and environment for statistical computing.

THABTAH, F., GHARAIBEH, O. & ABDELJABER, H. Comparison of rule based classification techniques for the Arabic textual data. Innovation in Information & Communication Technology (ISIICT), 2011 Fourth International Symposium on, 2011. 105-111.

TORUNOĞLU, D., ÇAKIRMAN, E., GANIZ, M. C., AKYOKUŞ, S. & GÜRBÜZ, M. Z. Analysis of preprocessing methods on classification of Turkish texts. Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on, 2011. IEEE, 112-117.

TU, S. & XU, L. 2012. A theoretical investigation of several model selection criteria for dimensionality reduction. *Pattern Recognition Letters,* 33**,** 1117-1126.

TWITTER. 2018a. *developer* [Online]. Available: https://developer.twitter.com/ [Accessed 23/03/2018].

TWITTER. 2018b. *Sample set of Twitter*

*glossary* [Online]. Available: https://help.twitter.com/en/glossary [Accessed 22/03/2018].

TWITTER. 2018c. *Twitter Documents* [Online]. Available: https://developer.twitter.com/en/docs [Accessed 22/03/2018].

TXTRAZOR. 2018. Available: https://www.textrazor.com/ [Accessed 22/03/2018].

VAPNIK, V. 2013. *The nature of statistical learning theory*, Springer science & business media.

WEIGEND, A. S., WIENER, E. D. & PEDERSEN, J. O. 1999. Exploiting hierarchy in text categorization. *Information Retrieval,* 1**,** 193-216.

WEISS, S. M., APTE, C., DAMERAU, F. J., JOHNSON, D. E., OLES, F. J., GOETZ, T. & HAMPP, T. 1999. Maximizing text-mining performance. *IEEE Intelligent Systems and their applications,* 14**,** 63-69.

WIENER, E., PEDERSEN, J. O. & WEIGEND, A. S. A neural network approach to topic spotting. Proceedings of SDAIR-95, 4th annual symposium on document analysis and information retrieval, 1995. Las Vegas, NV, 332.

WRIGHT, L. W., NARDINI, H. K. G., ARONSON, A. R. & RINDFLESCH, T. C. 1999. Hierarchical concept indexing of full-text documents in the Unified Medical Language System® Information Sources Map. *Journal of the American Society for Information Science,* 50**,** 514-523.

XUE, J. 2008. *Aspects of generative and discriminative classifiers.* University of Glasgow.

YAHIA, M. E. Arabic text categorization based on rough set classification. Computer Systems and Applications (AICCSA), 2011 9th IEEE/ACS International Conference on, 2011. IEEE, 293-294.

YANG, T.-H., YANG, S.-C., HO, J.-Y., LIN, G.-F., HWANG, G.-D. & LEE, C.-S. 2015. Flash flood warnings using the ensemble precipitation forecasting technique: A case study on forecasting floods in Taiwan caused by typhoons. *Journal of Hydrology,* 520**,** 367-378.

YANG, Y. A study of thresholding strategies for text categorization. Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 2001. ACM, 137-145.

YANG, Y. & PEDERSEN, J. O. A comparative study on feature selection in text categorization. Icml, 1997. 412-420.

YOUSSEF, A. M., SEFRY, S. A., PRADHAN, B. & ALFADAIL, E. A. 2016. Analysis on causes of flash flood in Jeddah city (Kingdom of Saudi Arabia) of 2009 and 2011 using multi-sensor remote sensing data and GIS. *Geomatics, Natural Hazards and Risk,* 7**,** 1018-1042.

ZANGERLE, E. & SPECHT, G. Sorry, I was hacked: a classification of compromised twitter accounts. Proceedings of the 29th Annual ACM Symposium on Applied Computing, 2014. ACM, 587-593.

ZAYED, O. H. & EL-BELTAGY, S. R. Person name extraction from modern standard Arabic or colloquial text. Informatics and Systems (INFOS), 2012 8th International Conference on, 2012. IEEE, NLP-44-NLP-48.

ZIRIKLY, A. & DIAB, M. 2014. Named entity recognition for dialectal arabic. *ANLP 2014***,** 78.

ZIRIKLY, A. & DIAB, M. Named entity recognition for arabic social media. Proceedings of naacl-hlt, 2015. 176-185.

ZRIGUI, M., AYADI, R., MARS, M. & MARAOUI, M. 2012. Arabic text classification framework based on latent dirichlet allocation. *CIT.Journal of Computing and Information Technology,* 20**,** 125-140.